ARPA ORDER NO.: 3486
9G10 Tactical Technology

LEVEL II (12)

R-2376-ARPA
January 1979

The Terrain Intervisibility and Movement Evaluation Routine (TIMER) Model

L. H. Wegner, M. G. Weiner

A Report prepared for

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

Rand
SANTA MONICA, CA. 90406

The research described in this report was sponsored by the Defense Advanced Research Projects Agency under Contract No. MDA903-78-C-0121.

Reports of The Rand Corporation do not necessarily reflect the opinions or policies of the sponsors of Rand research.

Library of Congress Cataloging in Publication Data

Wegner, L H
The Terrain Intervisibility and Movement Evaluation Routine (TIMER) model.

([Report] - Rand Corporation ; R-2376-ARPA)
1. TIMER (Computer program) 2. Tactics--Data processing. 3. War games--Data processing. I. Weiner, Milton Gershwin, 1923- joint author. II. United States. Defense Advanced Research Projects Agency. III. Title. IV. Series: Rand Corporation. Rand report ; R-2376-ARPA.
AS36.R3 R-2376 [U167] 081s [355.4'23'02854]
ISBN 0-8330-0083-7 78-27150

Published by The Rand Corporation

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

(14) RAND

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER R-2376-ARPA	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) The Terrain Intervisibility and Movement Evaluation Routine (TIMER) Model		5. TYPE OF REPORT & PERIOD COVERED (9) Interim rept.
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) (10) L. H. Wegner, M. G. Weiner		8. CONTRACT OR GRANT NUMBER(s) (15) MDA903-78-C-0121, VARPA Order-3486
9. PERFORMING ORGANIZATION NAME AND ADDRESS The Rand Corporation 1700 Main Street Santa Monica, California 90406		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS Defense Advanced Research Projects Agency Department of Defense Arlington, Virginia 22209		12. REPORT DATE (11) January 1979
		13. NUMBER OF PAGES 52
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) (12) 59p.		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for Public Release; Distribution Unlimited

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

No Restrictions

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Military Operations
Tactical Warfare
Terrain Models
Targets
Visibility
Computerized Simulation
War Games

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

See Reverse Side

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

296600

18

R-2376-ARPA The Terrain Intervisibility and Movement Evaluation Routine (TIMER) Model. J. H. Wegner, M. G. Weiner. January 1979.

This report is One of a series of three ~~reports~~ produced by a Rand study of the relationships between the tactical employment and the technological capabilities of advanced weapons for ground combat. The TIMER model was developed to determine the effects of terrain on target visibility during combat engagements. TIMER supplies data on the occurence and duration of target visibility from defense positions, the maximum range at which target acquisition might occur, and the potential rate at which a defense force could fire on an attacking force. The model uses the digitalized data base developed by the Defense Mapping Agency. The report includes the TIMER inputs and program listings, and illustrates the results that TIMER can produce. (See also R-2365, R-2377.) 59 pp. (WH)

1473

ARPA ORDER NO.: 3486
9G10 Tactical Technology

R-2376-ARPA
January 1979

The Terrain Intervisibility and Movement Evaluation Routine (TIMER) Model

L. H. Wegner, M. G. Weiner

A Report prepared for

DEFENSE ADVANCED RESEARCH PROJECTS AGENCY

Rand
SANTA MONICA, CA. 90406

PREFACE

This is one of a series of Rand reports documenting a study of advanced employment concepts for ground force operations, sponsored by the Tactical Technology Office of the Defense Advanced Research Projects Agency (ARPA). The overall objective of the study was to develop new concepts for the employment of ground combat systems incorporating advanced technology, and to define and evaluate weapon systems for implementing these concepts.

The first report, R-2365-ARPA, *A Method for Evaluating Advanced Systems and Concepts for Ground Combat*, by E. W. Paxson and M. G. Weiner, describes the general method of the study and discusses specific steps with examples from the evaluations carried out to date. The method incorporates a three-dimensional terrain board, a computer model, and a series of analytic modules programmed for a hand calculator to assess the outcomes of various combat engagements.

The present report describes in detail the computer model, TIMER, used in the evaluation method for determining the effects of terrain on visibility. It is being published separately for the convenience of readers who will find TIMER useful in applications other than the one described here.

A third volume, R-2377-ARPA, *Interactions Between Tactics and Technology in Ground Warfare*, by M. G. Weiner, E. W. Paxson, and R. A. Wise, presents a series of examples of interactions between tactics and technology that were observed in the combat evaluations conducted during the study. These examples form the basis for some general observations and speculations about tactical-technological interactions in future ground combat operations in Europe.

SUMMARY

The Terrain Intervisibility and Movement Evaluation Routine (TIMER) was developed as part of a Rand study of the relationships between the tactical employment and the technological capabilities of advanced weapon systems for ground combat. The purpose of TIMER was to determine the effects of terrain on target visibility during combat engagements carried out on a terrain board by human players. Among the questions that TIMER was designed to supply data on were: the occurrence and duration of target visibility (line of sight) from defense positions, the maximum range at which target acquisition might occur, and the potential rate at which a defense force could fire on ("service") an attacking force.

The TIMER model uses the digitalized terrain data base developed by the Defense Mapping Agency. With this fairly fine-grained data base, the program is able to determine whether there is a line of sight between any two points on the terrain. From this basic computation, a variety of conditions can be represented. It is possible to:

- Input one or more potential avenues or routes of attack and calculate the visibility of targets on the routes from various defense positions.
- Specify different target velocities and calculate the length of exposure of targets on the routes.
- Incorporate the reaction time of different defense systems and determine the number of opportunities to engage targets.
- Determine, for a given defense force occupying a number of different defense positions, their potential ability to "service" an attacker force approaching on one or several avenues of advance.

The report includes the TIMER inputs, program listings, and outputs for these conditions, and furnishes a number of illustrations of the results that TIMER can produce.

CONTENTS

PRECEDING PAGE BLANK

I. GENERAL DESCRIPTION OF TIMER

The Terrain Intervisibility and Movement Evaluation Routine (TIMER) is a computer model that was developed as part of a study of advanced employment concepts sponsored by the Tactical Technology Office of the Defense Advanced Research Projects Agency (ARPA). This study examined the relationships between the tactical employment and the technological capabilities of advanced weapon systems for ground combat.

A method for evaluating those systems was devised; it uses a three-dimensional terrain board, human players, and computer and analytic programs for conducting a minute-by-minute evaluation of a combat situation involving the advanced weapon systems. TIMER was developed to produce data on the effects of terrain on target visibility; among the questions it was designed to answer were: the occurrence and duration of target visibility (line of sight) from defense positions in specific terrain, the maximum range at which target acquisition might occur, and the potential rate at which a defense force could fire on ("service") an attacking force.

The current version of TIMER uses European terrain in the border area between the Federal Republic of Germany and the German Democratic Republic in the U.S. V Corps sector. An area of approximately 2000 km^2 is represented. The terrain data base for the area consists of digitalized data obtained from the Defense Mapping Agency, and includes a data point at every 12.5 meters in the area. For each data point, the elevation to the nearest meter is given, as well as whether the point is in a forest or urban area.*

With this fine-grained data base, the TIMER program permits the user to specify one or more defense position locations, down to the level of individual vehicles as well as one or more specific targets. The program determines whether there is a line of sight between each

---

*The Defense Mapping Agency digitalized data base for the Federal Republic of Germany covers a larger area than that used in the TIMER program, but the current program can be upgraded to a larger area.

defense position and the targets. From this basic computational capability, it is possible to:

- o Input one or more potential avenues or routes of attack and calculate the visibility of targets on these routes from a variety of different defense positions.
- o Specify different target velocities and determine the length of exposure of the targets on the routes.
- o Incorporate the reaction times of different defense systems and determine the number of opportunities to engage enemy targets.
- o Determine, for a given defense force occupying a number of different defense positions, their potential ability to "service" an attacker force approaching on one or several avenues of advance.

These and other conditions and situations can be incorporated in the model to provide quantitative data on the effects of terrain, vegetation, and urban build-up of a geographic area on the potential of combat systems.

II. INTERVISIBILITY CALCULATIONS

The fine-grained TIMER data base described in the Introduction and a basic point-to-point, line-of-sight calculation have been incorporated in the TIMER computer program. It determines intervisibility between defense positions and attacker vehicles moving along specified paths or routes. Appendix A presents a description of the data base and a program listing for the intervisibility computations.

The steps involved in using the program include:

- *Selection of routes or paths,* which requires detailed examination of topographic maps. Ancillary data not always found on maps may also be needed, such as the widths and load-bearing capacity of roads through forested areas. The selection of preferred routes must take into account not only trafficability, cover, and concealment afforded by the various topographic features, but also expected tactics employed by enemy mechanized units. Each route selected is defined for the computer as a sequence of points about 200 meters apart, and the route is taken to be linear between consecutive points.
- *Selection of defensive positions,* which also requires detailed examination of maps to take topographic details into account plus such factors as expected attack routes, number of defense positions, availability of withdrawal routes for the defenders, and characteristics of the defender and attacker weapons systems--maximum range, reaction time, velocity, and the like.

Figure 1 illustrates a set of attack routes and defense positions selected in a study involving two advanced systems, a laser beam-rider missile system for direct fire and an indirect fire weapon system that acquired targets by means of an elevated sensor. The units using these systems occupied defense positions, indicated

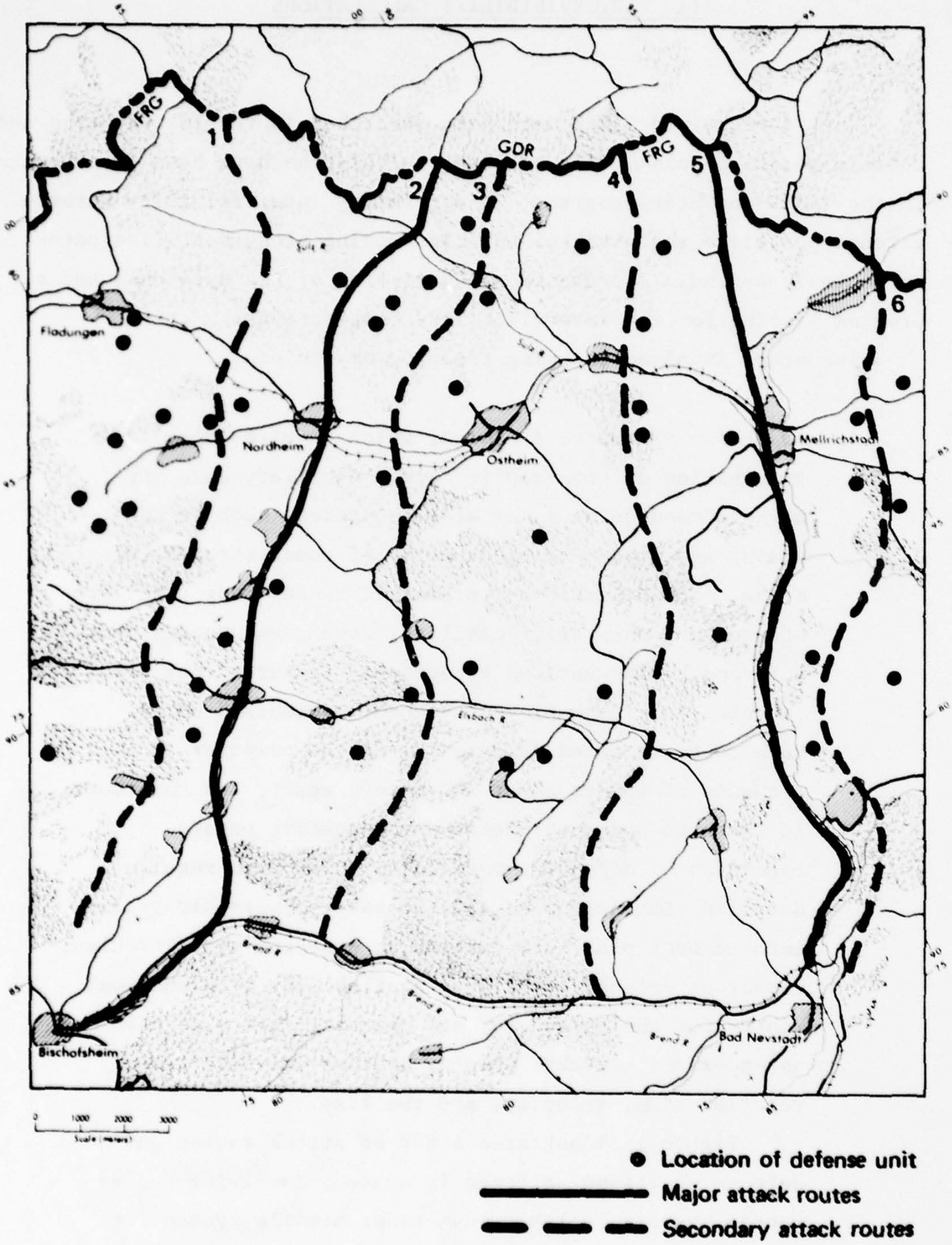

Fig. 1 — Initial defense force deployment and enemy attack routes

by dots in Fig. 1, to provide initial coverage of the likely (but unknown) routes of enemy advance. In one hypothetical situation, the enemy attacked with two regiments forward, and each regiment fanned out to place one battalion on each of three routes. The heavy solid lines in the figure indicate the two major attack routes, primarily using roads; the heavy dashed lines indicate four secondary attack routes, two flanking each major attack route.

The coordinates of each defense location are input to the computer program along with the following parameters that further characterize or constrain the capability of the individual observers:

Ranges of observers/sensors (maximum and minimum, km)
Coverage sector (angles measured from north)
Observer height (meters)
Target height (meters)
Forest and urban height (meters)

The computer program determines intervisibility between each defense position and each attack route for points along the route (at the input height above the route) a fixed increment apart (e.g., 25 m, an input parameter) taking into account the minimum and maximum range, coverage sector, observer and target heights, and forest and urban heights. Table 1 illustrates the TIMER output for a single position on a single attack route. The table presents the route segment intervisibility in a condensed form.*

The input values for the TIMER run are listed at the top of the table. The coordinates of the observation point are 65.20 km Easting and 121.70 km Northing, and the viewing sector is between 30 and 150 degrees measured from north. The surface elevation at the observer location is obtained from the data base. The 0 value for "observer

---

*The output from TIMER also includes a disk file data set containing the range to each visible segment on each route.

Table 1

VISIBLE ROUTE STRETCHES

Input Parameters

Observer location:	X = 65.20 km Y = 121.70 km
Observer symbol:	E92
Route symbol:	C4
Viewing sector:	30 to 150 degrees clockwise from north
Minimum range:	0.75 km
Observer height:	30 m
Target height:	2 m
Forest height:	15 m
Town height:	10 m

Calculated Values

Observer location surface elevation:	278 m
Observer location topography:	0

Route Seg.	Xcoord	Ycoord	Dist	Bear	Maximum Range 0.75	1.50	3.0
101	66.49	22.60	1.6	55.1	0	0	1
103	66.44	22.59	1.5	54.3	0	0	3
125	65.94	22.41	1.0	45.3	0	1	1
135	65.72	22.27	0.8	42.5	0	11	11

location topography" indicates that the observer location is not in a forest or town.

The bottom part of the table shows the output. It lists the route segment, coordinates, distance, and bearing from the observer or sensor location at which intervisibility starts and ends for the three sensor ranges given. For example, at the maximum sensor range of 3.0 km, intervisibility first occurs at segment 101 (the 101st 25-m segment from the start of the route) and lasts to segment 103, a total of three segments, or a stretch of 75 m. Intervisibility does not occur again until segment 125 but then continues until segment 135, a total of 11 segments.

For a maximum sensor range of 1.50 km, intervisibility occurs only for segments 125 to 135, because segments 101 to 103 are beyond the

1.5 km range. And for a maximum sensor range of 0.75 km, no intervisibilities are calculated, because the minimum range specified in the inputs is 0.75 km.

A similar calculation is made from every observer or sensor location to every route within the viewing sector specified in the input. However, it does not present, in a conventional form, an integrated picture of the intervisibility from all of the observation points on all of the routes. This is obtained by combining a TIMER output data set containing the ranges to all of the visible segments with a modified TIMER run indicating which segments are not visible because they are in a forest or town. Figures 2 and 3 illustrate the results of this procedure. Figure 2 shows the stretches of the six attack routes in Fig. 1 that are in the "open" and the stretches that are in forests and towns. Only about half the total length of the routes is in the open, although individual routes vary from 22 percent to 82 percent open. The "open" areas represent *potentially* visible portions of the routes, but not all of them are actually visible from the defense positions, because hills, forests, and towns may interfere with the line of sight. Figure 3 shows the percentages of the routes that are actually visible forward from one or more of the defense positions, for a situation in which the sensors of the indirect fire systems are elevated to a height of 30 meters. About 30 percent of all the routes combined are "covered" by this elevated sensor, and about 50 percent of the potentially visible stretches are covered.

The examples presented in this section indicate how the TIMER model can be used to determine where and how much coverage of potential attack routes can be obtained from one or more defense positions with elevated sensors. Although not illustrated here, the coverage from each defense position is obtained, as well as the amount and location of overlapping coverage from other defense positions. Similar data can be obtained for defense positions that do not have elevated sensor platforms.

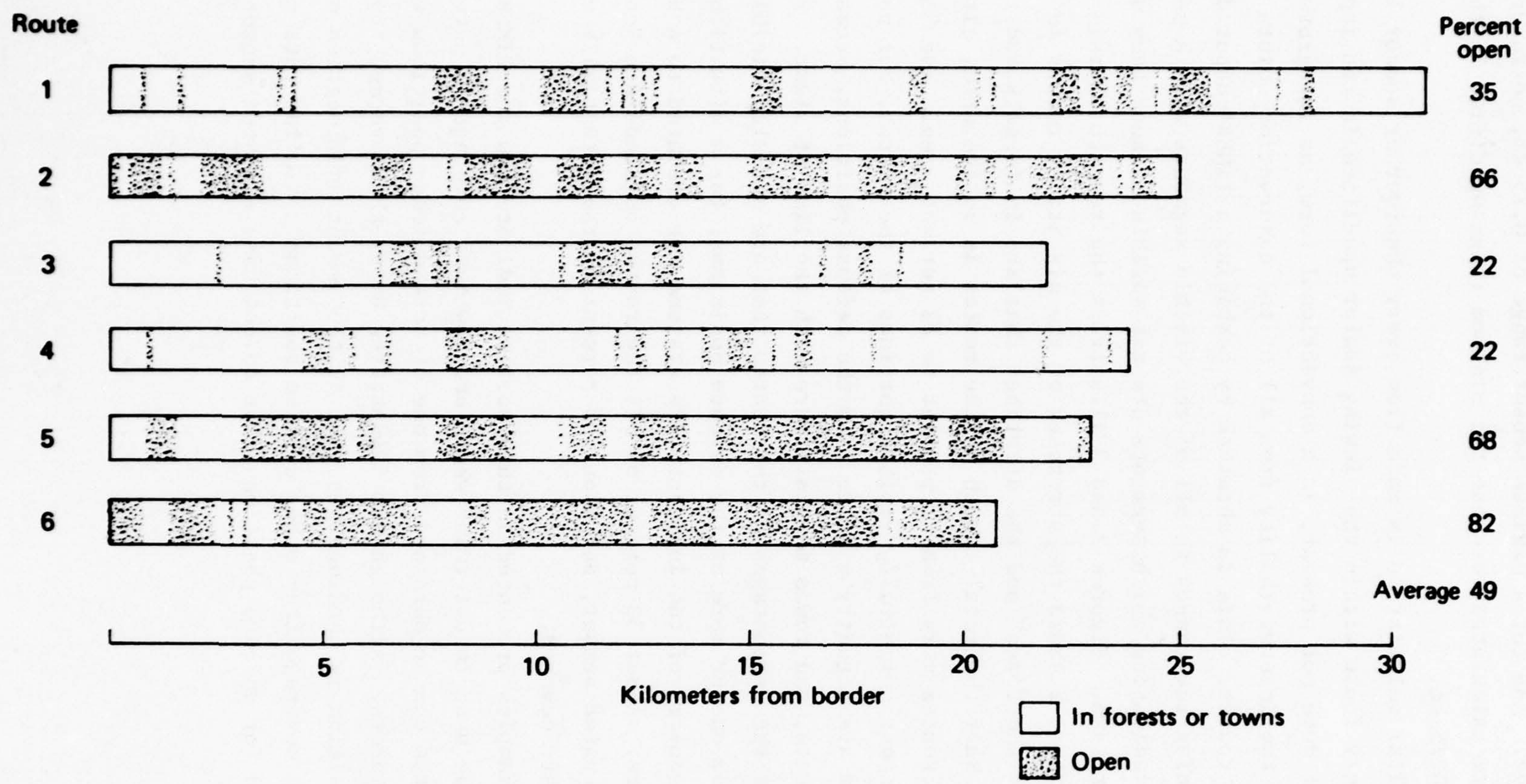

Fig. 2 — Open stretches of six attack routes

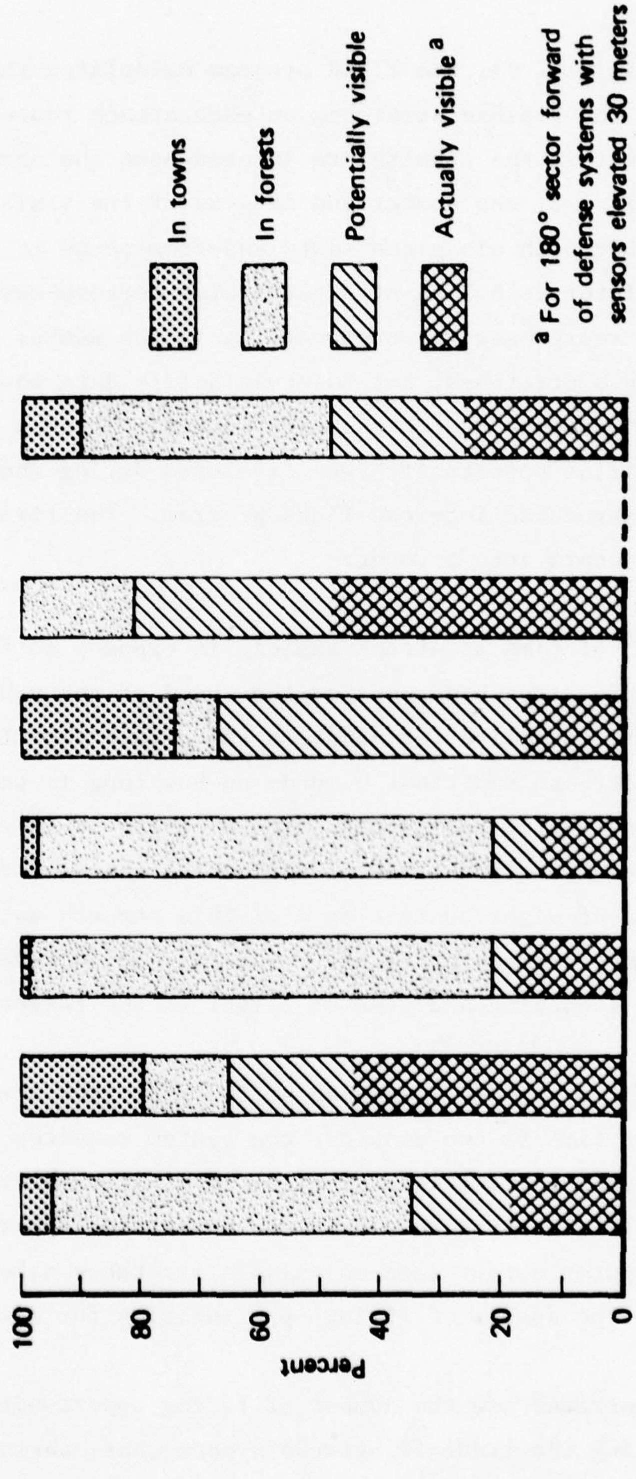

Fig. 3 — Percent of total route potentially and actually visible

III. FIRING OPPORTUNITY CALCULATIONS

As described in Sec. II, the TIMER program calculates the locations and lengths of all the visible stretches on each attack route for each defense position, within the constraints imposed upon the observer. As illustrated in Fig. 3, the number and lengths of the visible stretches can vary considerably both along the route and from route to route. The extent of the intervisibility of a particular defense-observer/route combination can be represented by such metrics as the number and average length of the visible stretches, but intervisibility data should be related directly to defense system characteristics. To this end, a metric called a "firing opportunity" was developed during the course of the study and introduced into the TIMER program. The firing opportunity takes two factors into account:

- The length of time an attack vehicle is exposed to fire on a visible stretch depends on the speed of the vehicle.
- The ability to deliver ordnance on the attack vehicle from the defense positions depends on how long it takes the defense system to respond, once a target is in view. This system "reaction time" is defined as the time from first line-of-sight contact on a visible stretch until the ordnance arrives at target. That is, it includes sighting, launching and time of flight to the target.

For example, if a vehicle is moving at 30 km/hr (500 m/min) and the system reaction time is two minutes, the system requires a visible stretch of 1000 m from initial line of sight to final ordnance delivery. Each 1000 m of visible stretch thus provides one firing opportunity. Processing the computer output data on visible stretches makes it possible to determine the number of firing opportunities for different conditions.

Figure 4 illustrates how the number of firing opportunities can be used in evaluating the tradeoff between system characteristics. In

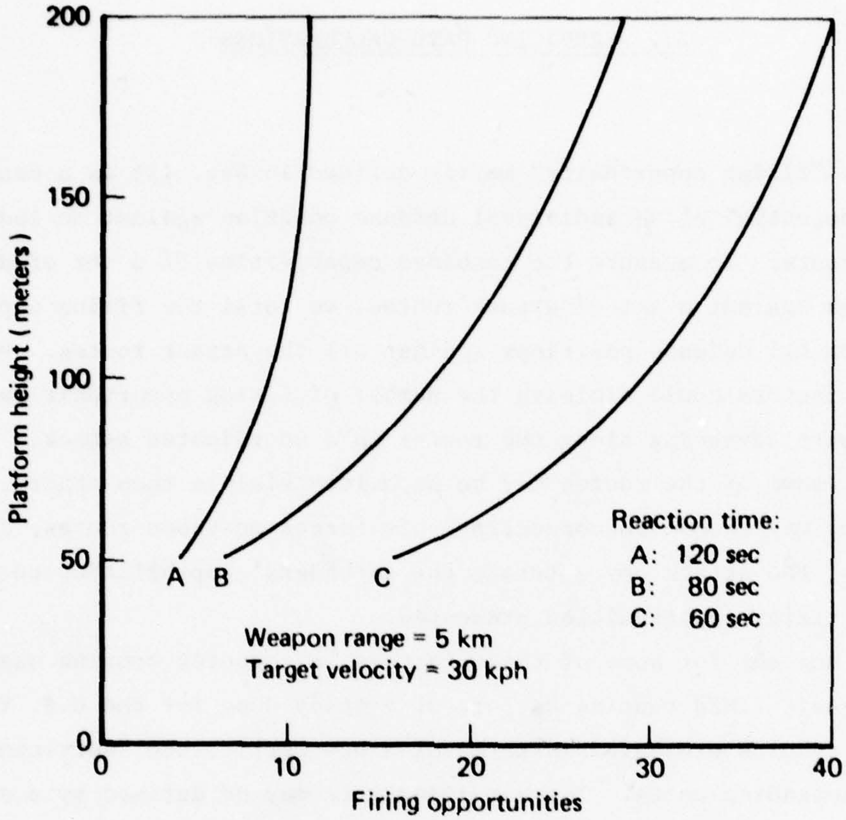

Fig. 4 — Guided-mortar firing opportunities for various reaction times

the figure, firing opportunities are given for an indirect fire system that has an elevated sensor platform for target acquisition. The figure shows total firing opportunities as a function of platform height and system reaction time. As expected, firing opportunities increase as the height of the sensor increases or as the system reaction time decreases. The results provide one basis for determining the value of either increasing the sensor height or decreasing the system reaction time, and also the tradeoff between the two. For example, if the nominal platform height is 50 m and the nominal reaction time is 120 sec, doubling the platform height to 100 m doubles the number of firing opportunities from 5 to 10, whereas halving the sytem reaction time from 120 sec to 60 sec more than triples the number of firing opportunities, from 5 to 16.

IV. SERVICING RATE CALCULATIONS

The "firing opportunity" metric defined in Sec. III is a measure of the potential of an individual defense position against an individual attack route. To measure the combined capabilities of a set of defense positions against a set of attack routes, we total the firing opportunities from all defense positions against all the attack routes. However, certain factors could diminish the number of firing opportunities on enemy units advancing along the routes in a coordinated attack. For example, some of the routes may be much less visible than others, and the enemy may choose to concentrate his forces on those routes, or the timing of the attack may saturate the defenders' capabilities to use all the firing opportunities presented.

To account for some of these factors, a computer routine was added to the basic TIMER routine as part of a study done for the U.S. V Corps and the results presented in terms of a new metric, the "servicing rate" of the defending units. The servicing rate may be defined by a simple example: A defending force that fires upon 15 tanks in five minutes will be said to have a servicing rate of three tanks per minute. Although the servicing rate of a given force is situation-dependent and the computer model is a simulation of a particular situation, the illustration presented below covers a number of aspects.

The situation is one in which enemy tank companies consisting of ten tanks each are attacking along each of ten routes with an approximate spacing of 50 m between tanks. All companies are moving at the same constant speed of 18 km/hr or 5 m/sec. The starting positions on the routes are inputs and may be arbitrary, but they may be chosen to meet some timing criterion--for example, each tank company arrives at a specified position along its route at a specified time.

The defending fire units need not be all of the same type. They might consist of a mixture of tanks, Sheridans, Tows, and Dragons positioned at locations in an integrated defensive posture. Different tactical procedures may be represented in the model. For example, each defense unit may remain in position or it may fire from its position

and then move to a nearby position to avoid return fire. Each weapon might have a specific number of rounds available. After the time taken to complete the move, the unit resumes fire as targets become visible. If two or more targets become visible simultaneously, the target with the smallest number of previously assigned rounds is chosen. This procedure approximates an integrated fire-coordination capability and, in effect, also serves as a proxy for removing targets that would have been destroyed by previous rounds.

The process of firing, moving, and occupying a new defense position is continued until the rounds available are exhausted, until some tank company approaches so close to the defense position that the defense position is abandoned, or until an enemy tank company reaches its objective.

Table 2 presents the computer input to the servicing rate program for this illustration. The maximum and minimum ranges given in the table are not the real range limits for existing weapon systems. The maximum ranges were selected to represent bad weather visibility conditions and the minimum ranges were selected to represent the range at which the defense units move out of their positions to avoid becoming overrun.

The output from the program, which is listed in App. B, is a minute-by-minute indication of events. Table 3 presents a sample output at Minute 8 of a computer run involving companies of tanks attacking

Table 2

EXAMPLE OF INPUTS TO SERVICING RATE ROUTINE

Velocity = 18 km/hr
Maximum rounds per target = 4

Input	Weapon 1	Weapon 2	Weapon 3	Weapon 4
No. of rounds available	40	29	10	6
Min firing range	750 m	750 m	750 m	750 m
Max firing range	1000 m	1000 m	1000 m	1000 m
Firing rate 1 per	30 sec	45 sec	45 sec	45 sec
Firing interval	60 sec	90 sec	90 sec	90 sec
Movement interval	120 sec	120 sec	120 sec	120 sec

Table 3

EXAMPLE SERVICING RATE ROUTINE OUTPUT

Cumulative results: Time = 8 Minutes

Route No.	Route Name	Attacker Tank Number 1	2	3	4	5	6	7	8	9	10	Total
1	C1	4	2	2	2	2	0	0	0	0	0	12
2	C2	1	0	0	1	0	0	0	0	1	0	3
3	C3	0	1	0	0	1	0	0	0	0	0	2
4	C4	1	0	0	1	0	0	0	0	0	0	2
5	C5	0	0	0	0	0	0	0	0	0	0	0
6	C6	2	2	0	1	1	0	0	0	0	0	6
7	C7	4	1	1	1	1	1	0	0	0	0	9
8	C8	1	0	1	0	0	0	0	0	0	0	2
9	C9	1	1	2	0	0	0	0	0	0	0	4
10	C10	0	0	0	0	0	0	0	0	0	0	0

Number of targets serviced through this minute = 27

Position No.	Position Name	Routes C1	C2	C3	C4	C5	C6	C7	C8	C9	C10	Tot.	Cum.
19	F71S	0	0	0	0	0	1	0	0	0	0	1	1
20	F72S	0	0	0	0	0	2	0	0	0	0	2	3
21	F73S	0	0	0	0	0	1	0	0	0	0	1	4
22	F74S	0	0	0	0	0	2	0	0	0	0	2	6
23	F75S	0	0	0	0	0	0	2	0	0	0	2	8
24	F76S	0	0	0	0	0	0	1	1	0	0	2	10
25	F81S	0	0	0	0	0	0	0	0	1	0	1	11
58	E44T	0	2	0	0	0	0	0	0	0	0	2	13
59	E45T	2	0	0	0	0	0	0	0	0	0	2	15
60	E61T	2	0	0	0	0	0	0	0	0	0	2	17
61	E62T	2	0	0	0	0	0	0	0	0	0	2	19
62	E63T	2	0	0	0	0	0	0	0	0	0	2	21
63	E64T	2	0	0	0	0	0	0	0	0	0	2	23
64	E65T	2	0	0	0	0	0	0	0	0	0	2	25
65	E91T	0	1	2	0	0	0	0	0	0	0	3	28
71	F22T	0	0	0	2	0	0	0	0	0	0	2	30
75	F31T	0	0	0	0	0	0	0	0	2	0	2	32
76	F32T	0	0	0	0	0	0	1	0	1	0	2	34
77	F33T	0	0	0	0	0	0	2	0	0	0	2	36
78	F34T	0	0	0	0	0	0	1	1	0	0	2	38
79	F35T	0	0	0	0	0	0	2	0	0	0	2	40
Total		12	3	2	2	0	6	9	2	4	0	40	

on each of ten routes. Output data in the top half of the table show the number of rounds that fired against each of the ten tanks on each of the routes through the eighth minute. On route C2, for example (second row of the table), enemy tanks numbered 1, 4, and 9 were fired upon once during the first eight minutes. The "Total" column tabulates the rounds fired on each route. During the eight minutes, 27 tanks (the number of cells in the body of the table with entries other than 0) have been "serviced."

The lower portion of the table presents the cumulative number of rounds fired from each defensive position, indicated by the identification code at the left, at all the tanks on each route. For example, defense weapon number 71, Code F22T, has fired two rounds against tanks on Route C4. A cross-comparison of the two parts of the table shows that the two rounds fired by this defense weapon were directed against tanks 1 and 4 of the tank company on that route. As indicated, 27 tanks have been "serviced" through the eighth minute at least once, and a total of 40 rounds have been dired by the defense units. From the output tables for each minute, the *entire* "servicing" picture can be constructed to determine which weapons fired how many rounds at which enemy vehicles at what time, taking into account the intervisibilities of the specific terrain.

The detailed data in Table 3 can be summarized to show the cumulative number of enemy vehicles serviced over a period of time as well as the number serviced each minute. Figure 5 presents an example of such a summary for a force of 100 enemy tanks attacking on ten different routes. The figure shows that *for the specific terrain* used in the analysis, the intervisibility conditions, combined with the dispositions and weapon capabilities of the defense force, provide this force with the *potential* to service all of the enemy tanks and that they are all serviced within 12 minutes after the start of the attack.

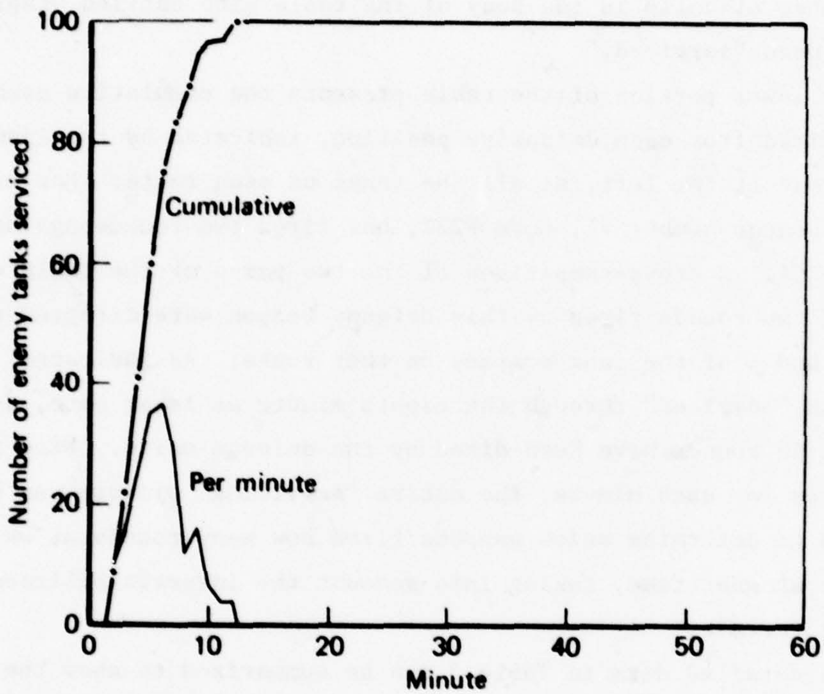

Fig. 5 — "Servicing" potential against an enemy force of 100 tanks

V. USES AND LIMITATIONS OF TIMER

USES OF TIMER

TIMER was developed for, and has been used in, several ARPA studies of advanced system concepts. In these studies, the model provided data on acquisition ranges, time and distance of target exposure, firing opportunities, and other factors for direct fire. TIMER has also been used in some terrain analyses of forward defense positions in U.S. V Corps. Data from TIMER have also contributed to several studies of airborne platforms.

In view of the emphasis being placed on new concepts of ground combat that take advantage of terrain ("terrain tuning"), rapid response ("management of seconds"), and precision guided munitions (PGMs), TIMER can be a useful tool for detailed assessment of the interplay among terrain, intervisibility of weapons, and movement of units.

LIMITATIONS OF TIMER

The TIMER model uses a digitalized terrain data base and a series of computer programs. The digitalized terrain data base is generally of good quality, but several characteristics should be recognized. Recent clearing of forests and new construction in urban areas have not been incorporated in the data base. No attempt has been made to determine the specific magnitude of these changes, but they appear to be unimportant thus far.

A second characteristic of the data base is that some local features--such as a small clump of trees, a single building, a small undulation in the terrain, a tree-lined road--are not included in the data base. These could influence intervisibility in some specific cases; however, for evaluations involving multiple defense positions and multiple attack routes or vehicles, their effects are considerably reduced. Another characteristic is that urban areas are treated as bounded areas: "Openings" such as central squares, plazas, wide roads, and soccer fields are not distinguished in the data base.

These characteristics influenced the degree of resolution that could be achieved in some of the specific computations made in the ARPA studies, but the overall quality of the data base makes it a unique and valuable tool for most intervisibility studies.

The current computer programs of TIMER have several limitations. They do not include a capability to determine the dynamic effects on reduced intervisibility due to local weather conditions, smoke, dust, etc., on target servicing rates. They also do not include the capability to vary the velocity of moving vehicles in response to undulations of the terrain or the surface conditions of roads and cross-country areas.

Finally, TIMER is not a two-sided engagement model. It does not incorporate hit probabilities, weapons effects, or exchanges of fire. In the evaluation method described in this series of reports, such factors were handled by constructing probabilistic Monte Carlo models and sampling by random numbers to produce realistic outcomes of combat engagements.

Appendix A

THE TIMER PROGRAM AND DATA BASE

The TIMER program takes inputs characterizing an observer and the locations defining the routes (the routes are linear between these locations), and determines the route segments (equal-length portions of the route starting from the first location) that are visible to the observer (see Table 1). Several types of output are constructed from these basic intervisibility data:

- A frequency table of the number of contiguous visible route segments as a function of sensor range;
- A frequency table of the number of firing opportunities as a function of the velocity of the target;
- A stylized map with a resolution of 100 m and containing a letter or other symbol indicating the location of the observer and a set of symbols indicating route segments in forest or town (*), the symbol of the observer for visible segments (the letter or other symbol), and a decimal point if the segment is not visible and not in a forest or town;
- An output "intervisibility" data set containing, for each observer/route combination, a table whose place positions represent the consecutive segments along the route and whose entries are the ranges from the observer to the route segments, if the segments are visible.

THE BASIC LINE-OF-SIGHT CALCULATION

The determination of a line of sight between two points on a map is conceptually straightforward: Simply draw a straight line between the two points and determine whether any point on the earth's surface (plus forest height or urban height if appropriate) in the plane determined by the line and the earth's center lies above the line. In TIMER, the following calculational procedure was implemented:

Let P1 = (x1,y1,h1) and P2 = (x2,y2,h2) be the Universal Transverse Mercator (UTM) coordinates and elevation above the earth's surface (m) of the observer and target locations, respectively. Let

$$d = \mathrm{sqrt}[(x1 - x2)^2 + (y1 + y2)^2]$$

be the (approximate) distance between the two points at the earth's surface. Next, let H1 and H2 be the elevation (m) of the two locations, as measured relative to the tangent plane at the observer location, so that H1 = h1 and

$$H2 = h2 - 500\ d^2/R,$$

where R is the earth's radius (km) and the expression subtracted from h2 is the approximate drop below the tangent plane due to the earth's curvature. Next, divide the distance d by the grid size (12.5 meters) to obtain the number of equally spaced intermediate points N between P1 and P2 at which the line of sight is to be tested. Each intermediate point is then tested by determining its elevation above the tangent plane (equal to its elevation obtained by interpolating in the data base minus the drop due to the earth's curvature between the observer location and the test point plus the forest or urban height if any) relative to the value obtained by interpolating between H1 and H2 on distance between P1 and the test point. If the test elevation is larger than the interpolated value, the line of sight is blocked at that point; if it is smaller, test the next point.

ORGANIZATION OF THE TIMER DATA BASE

The accurate determination of a line of sight between two points on or near the surface of the earth requires a fine-grained representation of the topography intervening between the two points. The Topographic Center of the Defense Mapping Agency has developed such a fine-grained data base for selected portions of Western Europe, primarily in the Federal Republic of Germany. The DMA data base consists of the elevation in meters and an indicator variable (= 1 if the point is in forest; = 2 if in a town; = 0 otherwise) for each data point on a 12.5 m grid for a number of rectangular regions in latitude and longitude; each rectangle encompasses an area 12' in latitude by 24' in longitude--i.e., 50° 12' to 50° 24' in latitude and 9° 12' to 9° 36' longitude.

The coordinate origin for each rectangular region is the intersection of the southernmost latitude and westernmost longitude, and the 12.5 m grid is in 12.5 m increments in UTM coordinates, Northing and Easting, starting from the origin.

To obtain a data base for an area contained in the DMA data base regions, the portions needed must be combined and the various origins kept track of, or data points must be interpolated for the grid points desired. For use in applications of TIMER, data bases for several sub-areas included within the overall DMA data base were prepared. The data base most often used, and referred to as the TIMER data base, is a portion in the border area between West and East Germany in the U.S. V Corps sector. A rectangular area (in UTM coordinates) approximately 2000 km^2 is represented, the region included between 5592-5654 km Northing and 547-579 km Easting in UTM zone 32.

The large number of data points (the data value at each data point is equal to ten times the elevation plus the value 0, 1, or 2 of the indicator variable) in the TIMER data base (3+ million) and the large number of elevation evaluations required to determine the existence of a line of sight between an observation point and a set of points defining a path or route, necessitated the adoption of a computationally efficient packaging of the data base for retrieval of elevation values. The TIMER data base grid of 12.5 m was arranged in 1 km by 1 km blocks of 1600 data points, starting with the southwest corner of the data and proceeding to the northwest corner and repeating from west to east.

The resulting data base of 62 (5654-5592 km Northing) x 32 (579-547 km Easting) = 1984 blocks of 1600 data points (or 3,174,400 total data points) were put on an IBM 2314 disk pack for permanent storage. The TIMER program was designed so that the data base permanent storage disk pack could be used directly, or any rectangular subregion of the TIMER data base could be written onto the IBM system disk packs and used by the program. (The latter procedure shortens turnaround time because the IBM 2314 disk pack does not have to be mounted for each computer run.) The computer program was written so that three of the 1 km by 1 km blocks of data points were always in the computer core memory:

the block containing the observation point, the block containing the current route point, and an intermediate block along the line of sight containing the current point at which elevation is to be determined (if not in the other two blocks). This procedure is a cost-effective tradeoff between computer core for data point storage and input operations for reading in the data points.

TIMER COMPUTATIONAL SEQUENCE

The TIMER program first reads an input card defining the map center and the number of 5 km by 12 km maps, in a north-south map strip, to be printed showing the routes. Then the program reads the data set defining the routes. The last "route" is not actually a route but is used to indicate the path of the border between the two Germanies on the "maps" that are printed containing the routes; this last route may be used to represent any other continuous descriptive feature. The maps of the routes are then printed.

Next, three cards are read defining parameter values that will be common to all the observers: the route segment length, the minimum visibility range, the number of map sections in the map indicating the observer/route intervisibility, the total number of observers, 11 maximum visibility ranges for defining the table of intervisibility versus range, and eight velocity values for defining the table of visibility time intervals versus velocity.

A card is then read for each observer containing the observer's coordinates, a four-character name for identifying the observer on output listings and maps, the heights of the observer and target, the heights of the forests and towns, and two angles defining the observer's coverage sector.

Starting with the first segment of the first route, the program determines the existence or nonexistence of a line of sight to each consecutive route segment. As the process proceeds, the program constructs an output table of first and last members of stretches of visible segments as a function of range (see Table 1). Using this table as a basis, two additional tables are constructed when the program reaches the end of a route: a frequency table of the number of

stretches of visible segments as a function of range, and a frequency table of intervisible time intervals as a function of velocity and range. The latter table can be interpreted as the number of firing opportunities given the firing rate. After all the routes are processed, the program creates an output "intervisibility" data set containing, for each observer/route combination, a one-dimensional table whose place positions represent the consecutive segments along the route and whose entries are the ranges from the observer to the segments if the segments are visible, and zero otherwise. The program prints a stylized map with a resolution of 100 meters and containing a letter or other symbol indicating the location of the observer and a set of symbols indicating route segments in forest or town(*), the observer symbol for visible route segments, and a decimal point if the segment is not visible and not in a forest or town.

TIMER INPUT FORMATS AND PROGRAM LISTING

Tables A.1, A.2, and A.3 present the TIMER input formats, the input format for the route data, and the program listing, respectively.

Table A.1

INPUT FORMATS--CARD DATA

Card Columns	Data Definitions
Card 1	
1-5	Number of map sections to be used for the map of the routes
6-10	X-coordinate (Easting) of the map center - 500 (km)
11-15	Y-coordinate (Northing) of the map center - 5500 (km)
16-20	Minimum X-coordinate (Easting) of the data base - 500 (km)
21-25	Maximum X-coordinate (Easting) of the data base - 500 (km)
26-30	Minimum Y-coordinate (Northing) of the data base - 5500 (km)
31-35	Maximum Y-coordinate (Northing) of the data base - 5500 (km)
36-40	Grid size (km)
Card 2	
1-10	Route segment length (m)
11-20	Minimum visibility rnage (m)
21-30	Number of map sections for map for each observer
31-40	Number of observers (for sizing output data set)
Card 3	
1-5	Maximum visibility range (km) number 1
...	
51-55	Maximum visibility range (km) number 11
Card 4	
1-5	Velocity (km/hr) number 1
...	
36-40	Velocity (km/hr) number 8
Card 5	
1-10	X-coordinate (Easting) - 500 (km) for the observer location
11-20	Y-coordinate (Northing) - 5500 (km) for the observer location
22-25	Four-character name for the observer
26-30	Height of the observer above the surface (m)
31-35	Viewing angle number 1 (degrees clockwise from north)
35-40	Viewing angle number 2 (degrees clockwise from north)
41-45	Height of the target (m)
46-50	Height of the forests (m)
51-55	Height of the towns (m)
56-60	Number of the observer (for position in output data set)
Card 6	
etc.	Same as Card 5 for additional observers

Table A.2

INPUT FORMAT--ROUTE DATA

The set of locations defining the routes is read from the route data set after the first input card. The data must be in order of location along the route. The order of the routes is arbitrary except that the last "route" is not actually a route but is the path defining the border (for printing on maps).

Card Columns	
7-10	Four-character name for the route
11-20	X-coordinate (Easting) - 500 (km) for the route location
21-30	Y-coordinate (Northing) - 5500 (km) for the route location

Table A.3

TIMER FORTRAN LISTING

```
C       MAIN
        COMMON/A/DPATH,RANGE(11),NSEG(25,2),XROUTE(2000),YROUTE(2000),
       1 DROUTE(2000),NQUAD(25,2),IQUADX(5000),IQUADY(5000),
       2 NOBS,NVQ1,NVQ,VQUADX(1000),VQUADY(1000),VTQUAD(1000),
       3 IROUTE(100),DLONG(100),TIME(100),VEL(100),
       4 HOBS,HTARG,HFOR,HTOWN,NROUTE,NMAP,R2MIN,R2MAX,
       5 SYMBOL(25,4),NAME(4),XLOW,XHIGH,YLOW,YHIGH,GSIZE,NVPMAX
        INTEGER*2 IQUADX,IQUADY,JQUADX,JQUADY,TQUAD,IROUTE
        INTEGER*2 VQUADX,VQUADY,VTQUAD,NAME,SYMBOL
        INTEGER*2 LENGTH(16),FTIME(16)
        INTEGER*2 MARKR(1600)
        DATA FTIME/5,10,20,30,40,60,75,100,120,150,180,240,300,360,480,
       * 600/
C
C TIMER PROGRAM FOR A PART OF GERMANY
C        NORTHING: 592-654 KM ; EASTING: 47-79 KM :  GSIZE=0.0125
C
        INTEGER*2 TF
        DATA TF/'*'/
        DEFINE FILE 3( 270,3200,L,JK)
C
        NOBS=0
        NVQ=0
C
C READ IN ROUTES AND LIST (LAST 'ROUTE' IS ACTUALLY BORDER)
C
        READ(5,98) NMAP,XCENT,YCENT,XLOW,XHIGH,YLOW,YHIGH,GSIZE
   98   FORMAT(I5,7F5.0)
        IF(XLOW.EQ.0.0)  XLOW  = 47.0
        IF(XHIGH.EQ.0.0) XHIGH = 79.0
        IF(YLOW.EQ.0.0)  YLOW  = 92.0
        IF(YHIGH.EQ.0.0) YHIGH = 154.0
        IF(GSIZE.EQ.0.0) GSIZE = 0.0125
        WRITE(6,99) NMAP,XCENT,YCENT,XLOW,XHIGH,YLOW,YHIGH,GSIZE
   99   FORMAT('1','NUMBER OF MAP PAGES FOR ROUTES: ',I3/
       * '0','MAP CENTER COORDINATES: X = ',F6.2,';  Y = ',F6.2/
       * '0','INPUT DATA REGION: EASTING = ',2F6.1,' NORTHING = ',2F6.1,
       * '   GSIZE = ',F6.4,' KM')
        CALL ROUTES
        CALL MAP(XCENT,YCENT,0,0)
C NR IS THE NUMBER OF ROUTES
        NR = NROUTE - 1
C
C READ CASE PARAMETERS
```

```
C
      READ(5,100) DPATH,RMIN,NMAP,NVPMAX,
     * RANGE,(VEL(I),I=1,8)
  100 FORMAT(2F10.0,2I10/11F5.0/8F5.0)
      WRITE(6,101) DPATH,RMIN,NMAP,NVPMAX,RANGE,(VEL(I),I=1,8)
  101 FORMAT('1',20X,'CASE PARAMETERS'//'0','ROUTE STEP SIZE =',
     * F5.1,' METERS'/'0','MINIMUM DETECTION RANGE: ',F10.0,' METERS'/
     * '0','NUMBER OF MAP PAGES FOR EACH OBSERVATION POINT:',I4/
     * '0','TOTAL OBSERVATION POINTS = ',I3/
     * '0','MAXIMUM DETECTION RANGES:', 11(1X,F6.2),'   (KM)'/'0',
     * 'VELOCITIES FOR EXPOSURE DISTRIBUTION:',8(1X,F5.1),'   (KM/HR)')
C
      R2MIN=RMIN*RMIN/(1000*1000)
C
C READ IN OBSERVER LOCATIONS
C
    1 READ(5,102,END=50) PX,PY,NAME,HOBS,ANGLE1,ANGLE2,HTARG,HFOR,
     * HTOWN,NVP
  102 FORMAT(2F10.0,1X,4A1,6F5.0,I10)
      NOBS=NOBS+1
C DETERMINE ELEVATION OF OBSERVATION POINT
      CALL ELEV(PX,PY,1,ALT,TOPOG)
C PLACE OBSERVATION POINT SYMBOL IN MAP TABLE
      NVQ = NVQ +1
      NVQ1 = NVQ
      I =10.0*(PX-XLOW+0.05)+1.001
      J =10.0*(PY-YLOW+0.05)+1.001
      VQUADX(NVQ) = I
      VQUADY(NVQ) = J
      VTQUAD(NVQ) = NAME(4)
      R2MAX=RANGE(11)**2
C DETERMINE OBSERVER SECTOR OUTLINE FOR MAP
      IF(ANGLE2-ANGLE1.NE.360.0)
     * CALL QUADS(PX,PY,ANGLE1,ANGLE2,NVQ)
      WRITE(6,104) PX,PY,ALT,TOPOG,NAME,HOBS,ANGLE1,ANGLE2,HTARG,
     * HFOR,HTOWN
  104 FORMAT('1','OBSERVATION POINT:  XCOORD =',F6.2,' YCOORD =',F6.2,
     * ' ALTITUDE =',F7.0,' TOPOG =',F3.0,'SYMBOL = ',4A1/
     * ' ','OBSERVER.HT. =',F5.0,
     * '  V.ANGLE 1 =',F6.0,'  V.ANGLE 2 =',F6.0,
     *   '  TARG.HT =',F6.1,'  FOR.HT =',F5.1,'  TOWN HT =',F5.1)
      WRITE(6,105) (FTIME(I),I=1,16)
  105 FORMAT('0',54X,'REQUIRED PATH LENGTH (METERS)'//
     1 ' ',52X,'         FIRING TIME (SEC.)'/
     2 ' ',25X,'  VEL.',7I4,9I5/' ',25X,'(KM/HR)')
      DO 3  I=1,8
      DO 2  J=1,16
    2 LENGTH(J) = (VEL(I)*FTIME(J))/3.6 + 0.5
    3 WRITE(6,106) VEL(I),(LENGTH(J),J=1,16)
  106 FORMAT(' ',25X,F5.0,1X,7I4,9I5)
```

```
C
C DETERMINE INTERVISIBILITY AND NUMBER OF FIRING OPPORTUNITIES
C FOR CURRENT OBSERVER
C
      DO 10 N = 1,NR
C ZERO OUT OUTPUT INTERVISIBILITY ARRAY
      DO 4 J=1,1600
    4 MARKR(J)=0
      NS1= NSEG(N,1)
      NS2= NSEG(N,2)
C TEST FOR LINE OF SIGHT TO SEGMENT NS OF ROUTE N
      DO 5  NS=NS1,NS2
      R2 = (PX-XROUTE(NS))**2 + (PY-YROUTE(NS))**2
C TEST IF SEGMENT IS WITHIN RANGE AND WITHIN VIEWING ANGLE
      IF(R2.GT.R2MAX) GO TO 5
      CALL VANGLE(NOBS,PX,PY,ANGLE1,ANGLE2,XROUTE(NS),YROUTE(NS),INOUT)
      IF(INOUT.EQ.1) GO TO 6
    5 CONTINUE
      GO TO 7
    6 WRITE(6,107) PX,PY,NAME,(SYMBOL(N,K),K=1,4)
  107 FORMAT('1','OBSERVATION POINT: X =',F6.2,' Y =',F6.2,' SYMBOL=',
     * 4A1,'  ROUTE :',4A1)
C DETERMINE FIRING OPPORTUNITIES ON ROUTE N FOR CURRENT OBSERVER
      CALL FIRE(N,PX,PY,ANGLE1,ANGLE2,MARKR)
    7 IF(NVP.EQ.0.OR.NVPMAX.EQ.0) GO TO 10
      JK=NVPMAX*(N-1)+NVP
      WRITE(3'JK) MARKR
   10 CONTINUE
C PRINT MAP SHOWING VISIBLE SEGMENTS ON ALL ROUTES
      WRITE(6,104) PX,PY,ALT,TOPOG,NAME,HOBS,ANGLE1,ANGLE2,HTARG,
     * HFOR,HTOWN
      CALL MAP(PX,PY,1,0)
      GO TO 1
   50 CALL EXIT
      END
```

```
      SUBROUTINE FIRE(NP,PX,PY,ANGLE1,ANGLE2,MARKR)
      COMMON/A/DPATH,RANGE(11),NSEG(25,2),XROUTE(2000),YROUTE(2000),
     1 DROUTE(2000),NQUAD(25,2),IQUADX(5000),IQUADY(5000),
     2 NOBS,NVQ1,NVQ,VQUADX(1000),VQUADY(1000),VTQUAD(1000),
     3 IROUTE(100),DLONG(100),TIME(100),VEL(100),
     4 HOBS,HTARG,HFOR,HTOWN,NROUTE,NMAP,R2MIN,R2MAX,
     5 SYMBOL(25,4),NAME(4),XLOW,XHIGH,YLOW,YHIGH,GSIZE,NVPMAX
      INTEGER*2 IQUADX,IQUADY,JQUADX,JQUADY,TQUAD,IROUTE
      INTEGER*2 VQUADX,VQUADY,VTQUAD,NAME,SYMBOL
      INTEGER*2 MARK(11),FIRING(11,160),IFIRE(31),TF,MARKR(1)
      DATA TF/'*'/
      DATA RADIAN/0.0174533/
C  NP=PATH NUMBER; NS=SEGMENT NUMBER; N=TEST POINT NUMBER
      WRITE(6,100) RANGE
  100 FORMAT('0',61X,'FIRING OPPORTUNITY SEGMENTS'/
     * ' ','ROUTE',63X,'MAX RANGE (KM)'/
     * ' ','POINT XCOORD YCOORD RANGE BEARING',
     * ' ', 11(F8.2)/)
      DO 1 I=1,11
      MARK(I) = 0
      DO 10 J=1,160
    1 FIRING(I,J) = 0
      IFLAG=0
      MARKP = 1
      DCUM = 0.0
      NPOINT = 1
      NS = NSEG(NP,1)
      X = XROUTE(NS)
      Y = YROUTE(NS)
    5 R2 = (X-PX)**2 + (Y-PY)**2
      INVIEW=1
      IF(R2.GT.R2MAX) INVIEW=0
      IF(R2.LT.R2MIN) INVIEW=0
      IF(INVIEW.EQ.0) GO TO 10
      CALL VANGLE(NVQ1,PX,PY,ANGLE1,ANGLE2,X,Y,INVIEW)
      IF(INVIEW.EQ.0) GO TO 10
      CALL LOS(NPOINT,PX,PY,X,Y,XT,YT,HTOPOG,INVIEW)
      IF(INVIEW.EQ.0.AND.(XT.NE.X.OR.YT.NE.Y)) GO TO 10
C SET MAP SYMBOL
      I = 10*(X-XLOW+0.05) + 1.001
      J = 10*(Y-YLOW+0.05) + 1.001
      II = VQUADX(NVQ)
      JJ = VQUADY(NVQ)
      IF(I.EQ.II.AND.J.EQ.JJ) GO TO 10
      NVQ = NVQ + 1
      IF(NVQ.GT.1000) WRITE(6,117)
  117 FORMAT('0','THE NUMBER OF VIEWED QUADS EXCEEDS THE MAX OF 1000')
      IF(NVQ.GT 1000) CALL EXIT
      VQUADX(NVQ) = I
      VQUADY(NVQ) = J
```

```
      VTQUAD(NVQ) = NAME(4)
      IF(INVIEW.EQ.1) GO TO 10
      VTQUAD(NVQ) = TF
C
C PRINT FIRING OPPORTUNITY SEGMENTS AND SET FIRING OPPORTUNITY TABLE
C
   10 IF(INVIEW.EQ.1.AND.NPOINT.LE.1600) MARKR(NPOINT)=1000*SQRT(R2)
      ISEE = 12
      IF(INVIEW.EQ.0) GO TO 12
      DO 11 I=1,11
   11 IF(RANGE(12-I)**2.GT.R2) ISEE=12-I
C ISEE IS NO. OF SMALLEST RANGE LARGER THAN DISTANCE TO TEST POINT
      IF(ISEE.EQ.1) GO TO 13
      IF(MARK(ISEE-1).EQ.0) GO TO 13
   12 IF(MARK(11).EQ.0) GO TO 13
C PRINT PREVIOUS TEST POINT AS END OF SEGMENT IF NOT ALREADY PRINTED
      IF(MARKP.EQ.1) GO TO 13
      R = SQRT(R2L)
C R2L IS SQUARED RANGE TO PREVIOUS TEST POINT
      TEMP1 = XL - PX
      TEMP2 = YL - PY
      BEAR = 90.0 - ATAN2(TEMP2,TEMP1)/RADIAN
      NN=NPOINT-1
      WRITE(6,101) NN,XL,YL,R,BEAR,(MARK(I),I=1,11)
C SET FIRING OPPORTUNITY TABLE
   13 DO 16 I=1,11
      IF(I.LT.ISEE) GO TO 14
      MARK(I) = MARK(I)+1
C TEST FOR LAST TEST POINT ON ROUTE
      IF(NS+1.EQ.NSEG(NP,2).AND.DCUM+DPATH/1000.0.GT.DROUTE(NS+1))
     * GO TO 14
      GO TO 16
   14 M = MARK(I)
      IF(M.EQ.0) GO TO 16
      M=MIN0(M,160)
      FIRING(I,M) = FIRING(I,M) + 1
      IFLAG=1
      MARK(I) = 0
   16 CONTINUE
C MARK CURRENT TEST POINT AS NOT PRINTED; IE NOT INITIAL SEGMENT POINT
      MARKP = 0
      IF(ISEE.EQ.12) GO TO 20
      IF(MARK(ISEE).NE.1) GO TO 20
C PRINT CURRENT TEST POINT AS INITIAL POINT OF A SEGMENT
   17 R = SQRT(R2)
      TEMP1 = X - PX
      TEMP2 = Y - PY
      BEAR = 90.0 - ATAN2(TEMP2,TEMP1)/RADIAN
      WRITE(6,101) NPOINT,X,Y,R,BEAR,(MARK(I),I=1,11)
  101 FORMAT(' ',I5,2F7.2,F6.1,F7.1,1X,11I8)
```

```
C MARK CURRENT TEST POINT AS PRINTED
      MARKP = 1
   20 DCUM = DCUM + DPATH/1000.0
   23 IF(DCUM.LE.DROUTE(NS+1)) GO TO 24
      NS = NS+1
      IF(NS.GE.NSEG(NP,2)) GO TO 30
      GO TO 23
   24 NPOINT = NPOINT+1
      XL=X
      YL=Y
      R2L=R2
      DEL =(DCUM-DROUTE(NS))/(DROUTE(NS+1)-DROUTE(NS))
      X = XROUTE(NS) + DEL*(XROUTE(NS+1)-XROUTE(NS))
      Y = YROUTE(NS) + DEL*(YROUTE(NS+1)-YROUTE(NS))
      GO TO 5
C
C PRINT FIRING OPPORTUNITY TABLE
C
   30 IF(IFLAG.EQ.0) RETURN
      WRITE(6,102) RANGE
  102 FORMAT('0',29X,'PATH',27X,'NUMBER OF FIRING OPPORTUNITIES'/
     * ' ',28X,'LENGTH',35X,'MAX RANGE (KM)'/
     1 ' ',26X,              'METERS)',11(F8.2)/)
      DO 40 J=1,160
      JJ = DPATH*J
      DO 41 I=1,11
      IFIRE(I)=0
      DO 35 JK=J,160
   35 IFIRE(I)=IFIRE(I)+FIRING(I,JK)*(JK/J)
      IF(IFIRE(11).EQ.0) GO TO 50
      WRITE(6,103) JJ,(IFIRE(I),I=1,11)
  103 FORMAT(' ',25X,I8,11I8)
   40 CONTINUE
C
C PRINT EXPOSURE TIME DISTRIBUTION
C
   50 WRITE(6,110) PX,PY,NAME,(SYMBOL(NP,I),I=1,4)
  110 FORMAT('1','OBSERVATION POINT: X =',F6.2,' Y =',F6.2,' SYMBOL=',
     * 4A1,' ROUTE :',4A1/'0',45X,'EXPOSURE TIME DISTRIBUTION'/
     *       ' ',45X,'   (20 SECOND INTERVALS)'/,
     * '0',' VEL  RANGE   0 20 40 60 80 10*12*14*16*18*20*22*24*26*',
     * '28*30*32*34*36*38*40*42*44*46*48*50*52*54*56*58*OVER TOTAL',
     * ' TOTAL  AVE   MAX'/
     * ' ',' KM/HR  KM    20 40 60 80 10*12*14*16*18*20*22*24*26*28*',
     * '30*32*34*36*38*40*42*44*46*48*50*52*54*56*58*60*600    #  ',
     * ' (SEC) (SEC) (SEC)')
      DO 60 IVEL=1,8
      IF(VEL(IVEL).EQ.0.0) GO TO 60
      IF(IVEL.EQ.5) WRITE(6,110) PX,PY,NAME,(SYMBOL(NP,I),I=1,4)
      WRITE(6,111)
```

```
111 FORMAT(' ')
    DO 60 IR=1,11
    XTIMET=0.0
    NTOTAL=0
    MAXT=0
    DO 54 K=1,31
 54 IFIRE(K)=0
    DO 55 J=1,160
    XTIME=J*DPATH*3.6/VEL(IVEL)
    K=XTIME/20.0+1
    IF(K.GT.30) K=31
    NFIRE=FIRING(IR,J)
    IFIRE(K)=IFIRE(K)+NFIRE
    XTIMET=XTIMET+XTIME*NFIRE
    NTOTAL=NTOTAL+NFIRE
    IF(NFIRE.NE.0) MAXT=XTIME+0.5
    ITIMET=XTIMET+0.5
 55 CONTINUE
    KAVE=XTIMET/MAX0(1,NTOTAL)+0.5
    WRITE(6,112) VEL(IVEL),RANGE(IR),(IFIRE(K),K=1,31),NTOTAL,
   * ITIMET,KAVE,MAXT
112 FORMAT(' ',F5.1,F6.2,2X,30I3,I4,4I6)
 60 CONTINUE
    RETURN
    END
```

```
      SUBROUTINE LOS(POINT,X1,Y1,X2,Y2,XF,YF,HTOPOG,LOS)
      COMMON/A/DPATH,RANGE(11),NSEG(25,2),XROUTE(2000),YROUTE(2000),
     1 DROUTE(2000),NQUAD(25,2),IQUADX(5000),IQUADY(5000),
     2 NOBS,NVQ1,NVQ,VQUADX(1000),VQUADY(1000),VTQUAD(1000),
     3 IROUTE(100),DLONG(100),TIME(100),VEL(100),
     4 HOBS,HTARG,HFOR,HTOWN,NROUTE,NMAP,R2MIN,R2MAX,
     5 SYMBOL(25,4),NAME(4),XLOW,XHIGH,YLOW,YHIGH,GSIZE,NVPMAX
      INTEGER*2 IQUADX,IQUADY,JQUADX,JQUADY,TQUAD,IROUTE
      INTEGER*2 VQUADX,VQUADY,VTQUAD,NAME,SYMBOL
      REAL*8 X,Y,H,DELX,DELY,DELH
C LOS = 1 INDICATES LINE OF SIGHT BETWEEN X1,Y1 AND X2,Y2
C LOS = 0 INDICATES NO LINE OF SIGHT
C XF,YF IS THE FIRST TEST POINT AT WHICH LINE OF SIGHT IS OBSCURED
C HTOPOG IS TOPOG AT XF,YF
      DATA RE/6378.388/
      LOS = 1
      DIST = SQRT((X1-X2)**2+(Y1-Y2)**2)
      IF(DIST.LE.GSIZE) RETURN
      CALL ELEV(X2,Y2,2,H2,TOPOG)
      H2 = H2 + HTARG
      IF(TOPOG.EQ.0.0)  GO TO 1
      IF(TOPOG.EQ.1.0.AND.HTARG.GT.HFOR) GO TO 1
      IF(TOPOG.EQ.2.0.AND.HTARG.GT.HTOWN) GO TO 1
      LOS = 0
      X = X2
      Y = Y2
      D = 0.0
      H = 0.0
      HACT = 0.0
      HTOPOG = HFOR
      IF(TOPOG.EQ.2.0) HTOPOG = HTOWN
      GO TO 6
    1 CALL ELEV(X1,Y1,1,H1,TOPOG)
      H1 = H1 + HOBS
      N = DIST/GSIZE
      DELX = (X2-X1)/(N+1)
      DELY = (Y2-Y1)/(N+1)
C CORRECT H2 FOR EARTH CURVATURE
      DELH = (H2-1000.0*0.5*DIST*DIST/RE - H1)/(N+1)
      DELD = DIST/(N+1)
      X = X1
      Y = Y1
      H = H1
      D = 0.0
C
      DO 5 M=1,N
      X = X + DELX
      Y = Y + DELY
      H = H + DELH
      D = D + DELD
```

```
      CALL ELEV(X,Y,3,HACT,TOPOG)
C CORRECT HACT FOR EARTH CURVATURE AND TEST FOR LINE OF SIGHT
      HACT = HACT - 1000.0*0.5*D*D/RE
      ITOPOG = TOPOG
      HTOPOG=0.0
      IF(ITOPOG.EQ.0) GO TO 4
      GO TO (2,3,2), ITOPOG
    2 HTOPOG = HFOR
      GO TO 4
    3 HTOPOG = HTOWN
    4 IF(H.GT.HACT+HTOPOG) GO TO 5
      LOS = 0
      GO TO 6
    5 CONTINUE
C
    6 XF=X
      YF=Y
      IF(HACT.GT.H) HTOPOG=0.0
  999 RETURN
      END
```

```
      SUBROUTINE ELEV(X,Y,IN,ALT,TOPOG)
      COMMON/A/DPATH,RANGE(11),NSEG(25,2),XROUTE(2000),YROUTE(2000),
     1 DROUTE(2000),NQUAD(25,2),IQUADX(5000),IQUADY(5000),
     2 NOBS,NVQ1,NVQ,VQUADX(1000),VQUADY(1000),VTQUAD(1000),
     3 IROUTE(100),DLONG(100),TIME(100),VEL(100),
     4 HOBS,HTARG,HFOR,HTOWN,NROUTE,NMAP,R2MIN,R2MAX,
     5 SYMBOL(25,4),NAME(4),XLOW,XHIGH,YLOW,YHIGH,GSIZE,NVPMAX
      INTEGER*2 IQUADX,IQUADY,JQUADX,JQUADY,TQUAD,IROUTE
      INTEGER*2 VQUADX,VQUADY,VTQUAD,NAME,SYMBOL
C
C ELEVATION IN METERS AND TOPOGRAPHY FOR POINTS IN A SELECTED PART OF
C GERMANY.  TOPOGRAPHY: 1=FOREST; 2=URBAN; 0=NEITHER FOREST NOR URBAN
C       NORTHING: 592-654 KM ; EASTING: 47-79 KM
C     X = EASTING(KM); Y = NORTHING(KM); ALT = ELEVATION(M)
C
      INTEGER*2 Z(40,40,3),Z1(1600),Z2(1600),Z3(1600)
      EQUIVALENCE (Z(1,1,1),Z1(1)),(Z(1,1,2),Z2(1)),(Z(1,1,3),Z3(1))
      INTEGER H(2,2).IH(2),IJG(3)
      REAL HY(2),XORG(8),YORG(8)
      DATA XORG/  0.002,  0.009,  0.004,  0.010,
     *            0.011,  0.011,  0.010,  0.008/
      DATA YORG/  0.004,  0.006,  0.008, 0.012,
     *            0.010,  0.011,  0.001,  0.004/
      DATA IJG/3*0/
      DEFINE FILE 1(7936,3200,L,IJ)
C
C THE PROGRAM FROM HERE TO FORTRAN STATEMENT 12 IS AN AD HOC WAY TO
C FIND THE ORIGIN CORRECTIONS FOR THE VARIOUS PORTIONS OF THE DATA BASE
C IT WOULD NOT BE NEEDED FOR OTHER DATA BASES
C
      IF(Y.LT.YLOW+0.0125.OR.Y.GT.YHIGH-0.0125) GO TO 13
      IF(X.LT.XLOW+0.0125.OR.X.GT.XHIGH-0.0125) GO TO 13
      IF(X.GE.71.0) GO TO 1
      IF(X.LE.70.0) GO TO 2
      D = 0.0135254*(Y-628.176) + 0.9999085*(X-70.473)
      IF(ABS(D).LE.0.0125) GO TO 12
      IF(D.LT.0.0) GO TO 2
    1 D = (0.999868*(Y-628.176)+0.016228*(X-70.473))/22.242 + 7.0
      GO TO 3
    2 D = (0.999942*(Y-628.176)+0.010799*(X-70.473))/22.242 + 3.0
    3 ID=D
      IF(D-ID.LE.0.001124) GO TO 12
      IF(D-ID.GE.0.998876) GO TO 12
      XOR= XLOW+XORG(ID)
      YOR= YLOW+YORG(ID)
      IFLAG=0
      GO TO 20
   12 IFLAG=1
      XOR= XLOW+GSIZE
      YOR= YLOW+GSIZE
```

```
      GO TO 20
   13 ALT=0.0
      TOPOG=0.0
      RETURN
C
   20 XI = (X-XOR)/GSIZE
       I = XI
      YJ = (Y-YOR)/GSIZE
       J = YJ
      DX = XI-I
      DY = YJ-J
      DO 30 II=1,2
      DO 29 JJ=1,2
      IND = IN
      IG = (I+II-1)/40
      JG = (J+JJ-1)/40
      IJ = 2*(YHIGH-YLOW)*IG+JG+1
      DO 21 K=1,3
      IF(IJ.NE.IJG(K)) GO TO 21
      IND = K
      GO TO 26
   21 CONTINUE
      GO TO (22,23,24), IND
   22 READ(1'IJ) Z1
      GO TO 25
   23 READ(1'IJ) Z2
      GO TO 25
   24 READ(1'IJ) Z3
   25 IJG(IND) = IJ-1
   26 H(II,JJ) = Z(I+II-40*IG,J+JJ-40*JG,IND)
      IF(IFLAG.EQ.1) GO TO 40
   29 IH(JJ) = H(II,JJ)/10
      HY(II) = (IH(2)-IH(1))*DY + IH(1)
   30 CONTINUE
      ALT = (HY(2)-HY(1))*DX + HY(1)
      I = 2*DX + 1
      J = 2*DY + 1
      TOPOG = H(I,J) - 10*(H(I,J)/10)
      RETURN
   40 ALT = H(1,1)/10
      TOPOG = H(1,1)-10*ALT
      RETURN
      END
```

```
      SUBROUTINE MAP(XCENT,YCENT,MODE,MTIME)
      COMMON/A/DPATH,RANGE(11),NSEG(25,2),XROUTE(2000),YROUTE(2000),
     1 DROUTE(2000),NQUAD(25,2),IQUADX(5000),IQUADY(5000),
     2 NOBS,NVQ1,NVQ,VQUADX(1000),VQUADY(1000),VTQUAD(1000),
     3 IROUTE(100),DLONG(100),TIME(100),VEL(100),
     4 HOBS,HTARG,HFOR,HTOWN,NROUTE,NMAP,R2MIN,R2MAX,
     5 SYMBOL(25,4),NAME(4),XLOW,XHIGH,YLOW,YHIGH,GSIZE,NVPMAX
      INTEGER*2 IQUADX,IQUADY,JQUADX,JQUADY,TQUAD,IROUTE
      INTEGER*2 VQUADX,VQUADY,VTQUAD,NAME,SYMBOL
      DIMENSION HEAD(10)
      INTEGER*2 A(121),B(26),P,TF,OH,BORDER,BLANK
      DATA TF/'*'/
      DATA P/'.'/,OH/'O'/,BORDER/'Z'/,BLANK/'  '/
      DATA HEAD/0.0,0.1,0.2,0.3,0.4,0.5,0.6,0.7,0.8,0.9/
C
C  PRINTS MAPS CENTERED AT XCENT,YCENT
C
      IMIN = 10*AINT(XCENT-5.0-XLOW)+1
      JMIN = 10*AINT(YCENT-(5*NMAP)/2-YLOW)+1
      I1 = XCENT-5.0
      I2 = I1+12
      DO 1 I=I1,I2
      II=I-I1+1
      B(II)=I/10
      B(II+13)=I-10*(I/10)
    1 CONTINUE
C
      IF(MTIME.NE.0) WRITE(6,100) MTIME
      IF(MTIME.EQ.0.AND.MODE.EQ.0) WRITE(6,104)
  100 FORMAT('1','TIME =',I4)
      WRITE(6,105) B
      NLINE=50*NMAP+1
      DO 20 J=1,NLINE
      JJ =  NLINE-J+1
      DO 11 I=1,121
   11 A(I) = BLANK
      DO 12 N=1,NROUTE
      K1 =NQUAD(N,1)
      K2 =NQUAD(N,2)
      DO 12 K=K1,K2
      IF(IQUADY(K).NE.JJ+JMIN-1) GO TO 12
      IF(IQUADX(K).LT.IMIN) GO TO 12
      IF(IQUADX(K).GT.IMIN+120) GO TO 12
      II = IQUADX(K)-IMIN+1
      A(II) = P
      IF(MTIME.EQ.0.AND.MODE.EQ.0) A(II) = SYMBOL(N,4)
      IF(N.EQ.NROUTE) A(II) = BORDER
   12 CONTINUE
C
   14 IF(NVQ.EQ.0) GO TO 17
```

```
      M1 = NVQ1
      M2 = NVQ
      IF(MODE.EQ.0) M1=1
      DO 16 M=M1,M2
      IF(VQUADY(M).NE.JJ+JMIN-1) GO TO 16
      IF(VQUADX(M).LT.IMIN) GO TO 16
      IF(VQUADX(M).GT.IMIN+120) GO TO 16
      II=VQUADX(M)-IMIN+1
      IF(MTIME.EQ.0) GO TO 15
      IF(A(II).EQ.P) GO TO 16
      IF(VTQUAD(M).EQ.TF) GO TO 16
      A(II) = OH
      GO TO 16
   15 A(II) = VTQUAD(M)
   16 CONTINUE
C
   17 JJJ = MOD(JJ-1,10) + 1
      JH = AINT(YCENT-(5*NMAP)/2) + (JJ-1)/10
      YH=JH
      H = HEAD(JJJ)
      IF(JJJ.EQ.1) WRITE(6,101) YH,A,YH
  101 FORMAT(' ',F6.1,121A1,F5.1)
      IF(JJJ.NE.1) WRITE(6,102) H,A,H
  102 FORMAT(' ',4X,F2.1,121A1,3X,F2.1)
      IF(JJ.EQ.1) WRITE(6,103) B
  103 FORMAT(' ',6X,12(' 123456789')/' ',6X,12(I1,'.........'),I1/
     * ' ',6X,12(I1,9X),I1)
      IF(MOD(JJ,50).NE.2.OR.JJ.EQ.2) GO TO 20
      WRITE(6,103) B
      WRITE(6,104)
  104 FORMAT('1'/)
      WRITE(6,105) B
  105 FORMAT(' ',6X,12(I1,9X),I1/,' ',6X,12(I1,'.........'),I1/
     * ' ',6X,12(' 123456789'))
   20 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE QUADS(X1,Y1,X2,Y2,CUMDIS,J)
      COMMON/A/DPATH,RANGE(11),NSEG(25,2),XROUTE(2000),YROUTE(2000),
     1 DROUTE(2000),NQUAD(25,2),IQUADX(5000),IQUADY(5000),
     2 NOBS,NVQ1,NVQ,VQUADX(1000),VQUADY(1000),VTQUAD(1000),
     3 IROUTE(100),DLONG(100),TIME(100),VEL(100),
     4 HOBS,HTARG,HFOR,HTOWN,NROUTE,NMAP,R2MIN,R2MAX,
     5 SYMBOL(25,4),NAME(4),XLOW,XHIGH,YLOW,YHIGH,GSIZE,NVPMAX
      INTEGER*2 IQUADX,IQUADY,JQUADX,JQUADY,TQUAD,IROUTE
      INTEGER*2 VQUADX,VQUADY,VTQUAD,NAME,SYMBOL
C DETERMINES QUADS ENTERED BY THE LINE SEGMENT FROM X1,Y1 TO X2,Y2
C FOR USE WITH MAP
C CUMDIS IS THE CUMULATIVE DISTANCE ALONG THE ROUTE
C J IS CURRENT NUMBER FOR QUAD TABLE
      X =10.0*(X1-XLOW+0.05)+1.001
      Y =10.0*(Y1-YLOW+0.05)+1.001
      DIST = 10.0*CUMDIS
      IX = X
      IY = Y
      IF(CUMDIS.NE.0.0) GO TO 10
      J=J+1
      IF(J.GT.5000) WRITE(6,100)
      IF(J.GT.5000) CALL EXIT
      IQUADX(J) = IX
      IQUADY(J) = IY
   10 DELX = 10*(X2-X1)
      DELY = 10*(Y2-Y1)
      D = SQRT(DELX**2 + DELY**2)
      IF(DELX.EQ.0.0) TX = 100.0
      IF(DELX.EQ.0.0) GO TO 1
      INCX = 1
      DELTX = ABS(1.0/DELX)
      FRACX = X - IX
      TX = (1.0 - FRACX)/DELX
      IF(DELX.GT.0.0) GO TO 1
      INCX = -1
      TX = -FRACX/DELX
    1 IF(DELY.EQ.0.0) TY = 100.0
      IF(DELY.EQ.0.0) GO TO 2
      INCY = 1
      DELTY = ABS(1.0/DELY)
      FRACY = Y - IY
      TY = (1.0 - FRACY)/DELY
      IF(DELY.GT.0.0) GO TO 2
      INCY = -1
      TY = -FRACY/DELY
    2 T = AMIN1(TX,TY)
      IF(T.GT.1.0) GO TO 5
      J = J + 1
      IF(J.LE.5000) GO TO 6
      WRITE(6,100)
```

```
100 FORMAT('0','THE NUMBER OF QUADS EXCEEDS THE MAX OF 5000')
    CALL EXIT
  6 IF(TX.GT.TY + 0.001) GO TO 3
    IX = IX +INCX
    TX = TX +DELTX
    IF(TY.LT.TX -DELTX + 0.001) GO TO 3
    GO TO 4
  3 IY = IY + INCY
    TY = TY + DELTY
  4 IQUADX(J) = IX
    IQUADY(J) = IY
    GO TO 2
  5 CUMDIS = CUMDIS + D/10.0
    RETURN
    END
```

```
      SUBROUTINE ROUTES
      COMMON/A/DPATH,RANGE(11),NSEG(25,2),XROUTE(2000),YROUTE(2000),
     1 DROUTE(2000),NQUAD(25,2),IQUADX(5000),IQUADY(5000),
     2 NOBS,NVQ1,NVQ,VQUADX(1000),VQUADY(1000),VTQUAD(1000),
     3 IROUTE(100),DLONG(100),TIME(100),VEL(100),
     4 HOBS,HTARG,HFOR,HTOWN,NROUTE,NMAP,R2MIN,R2MAX,
     5 SYMBOL(25,4),NAME(4),XLOW,XHIGH,YLOW,YHIGH,GSIZE,NVPMAX
      INTEGER*2 IQUADX,IQUADY,JQUADX,JQUADY,TQUAD,IROUTE
      INTEGER*2 VQUADX,VQUADY,VTQUAD,NAME,SYMBOL
      INTEGER*2 SYM(4),STEMP(4)
C
C READS IN ROUTE SEGMENTS AND LISTS
C ALSO DETERMINES QUADS ENTERED BY THE ROUTES (FOR MAP)
C
      NROUTE = 0
      DO 20 I=1,4
   20 STEMP(I)=0.0
C READ IN ROUTE SEGMENTS
      J=0
      NSEG(1,1)=1
    1 READ(2,100,END=10) SYM,X,Y
  100 FORMAT(6X,4A1,2F10.2)
      J = J + 1
      IF(J.GT.2000) WRITE(6,101)
  101 FORMAT('0','THE NUMBER OF ROUTE POINTS EXCEEDS THE MAX OF 2000')
      IF(J.GT.2000) CALL EXIT
      XROUTE(J) = X
      YROUTE(J) = Y
      DO 2 I=1,4
      IF(SYM(I).NE.STEMP(I)) GO TO 3
    2 CONTINUE
      NSEG(NROUTE,2)=NSEG(NROUTE,2)+1
      GO TO 1
    3 NROUTE = NROUTE+1
      DO 4 I=1,4
      SYMBOL(NROUTE,I) = SYM(I)
    4 STEMP(I) = SYM(I)
      IF(NROUTE.NE.1) NSEG(NROUTE,1)=NSEG(NROUTE-1,2)+1
      NSEG(NROUTE,2)=NSEG(NROUTE,1)
      GO TO 1
C PRINT ROUTE SEGMENTS
   10 WRITE(6,102)
  102 FORMAT('1',55X,'ROUTES'/'0','ROUTE',5X,5('   SEG. XCOORD YCOORD'))
      DO 12 N=1,NROUTE
      WRITE(6,103)
      NN = NSEG(N,1)
      NNN = MINO(NSEG(N,1)+4,NSEG(N,2))
   11 WRITE(6,103) (SYMBOL(N,I),I=1,4),
     * (M,XROUTE(M),YROUTE(M),M=NN,NNN)
  103 FORMAT(' ',4A1,6X,5(I7,F7.2,F7.2))
```

```
      NN = NN+5
      NNN = MINO(NNN+5,NSEG(N,2))
      IF(NN.LE.NSEG(N,2)) GO TO 11
   12 CONTINUE
C DETERMINE QUADS ENTERED BY ROUTES (FOR MAP)
      K=0
      NQUAD(1,1)=1
      DO 15 N=1,NROUTE
      IF(N.NE.1) NQUAD(N,1)=NQUAD(N-1,2)+1
      NQUAD(N,2)=NQUAD(N,1)
      JJ  = NSEG(N,1)
      JJJ = NSEG(N,2)
      DROUTE(JJ) = 0.0
      CUMDIS = 0.0
      DO 15 J=JJ,JJJ
      X = XROUTE(J)
      Y = YROUTE(J)
      IF(J.EQ.JJ) GO TO 15
      X1=XROUTE(J-1)
      Y1=YROUTE(J-1)
      CALL QUADS(X1,Y1,X,Y,CUMDIS,K)
      DROUTE(J)=CUMDIS
      NQUAD(N,2)=K
   15 CONTINUE
      REWIND 2
      RETURN
      END
```

```
      SUBROUTINE VANGLE(N,X,Y,ANGLE1,ANGLE2,XT,YT,INOUT)
C DETERMINES IF POINT XT,YT IS WITHIN VIEWING ANGLE FROM X,Y
C VIEWING ANGLE IS FROM ANGLE1 TO ANGLE2 IN CLOCKWISE DIRECTION
C BOTH ANGLES ARE MEASURED FROM NORTH DIRECTION ON GRID
C INOUT = 0 IF POINT XT,YT IS NOT WITHIN VIEWING ANGLE
C INOUT = 1 IF POINT XT,YT IS WITHIN TH VIEWING ANGLE
      DATA RADIAN/0.0174533/,NP/0/
      IF(NP.EQ.N) GO TO 5
      NP=N
      PX=X
      PY=Y
      A1=ANGLE1
      A2=ANGLE2
      IF(A1.GE.A2) A2=A2+360.0
C SET UP VECTORS FOR VIEWING ANGLE
      X1 = COS((90-A1)*RADIAN)
      Y1 = SIN((90-A1)*RADIAN)
      X2 = COS((90-A2)*RADIAN)
      Y2 = SIN((90-A2)*RADIAN)
C DETERMINE IF POINT IS WITHIN VIEWING ANGLE (BY PARALLEL PROJECTIONS)
    5 IF(A2-A1.NE.360.0) GO TO 1
      INOUT=1
      RETURN
    1 TEMP1 = (XT-PX)*Y1 - (YT-PY)*X1
      TEMP2 = (XT-PX)*Y2 - (YT-PY)*X2
      TEMP3 = X1*Y2 - Y1*X2
      IF(180.0-(A2-A1)) 2,3,4
    2 INOUT = 0
      IF(TEMP1/TEMP3.GT.0.0) INOUT=1
      IF(TEMP2/TEMP3.LT.0.0) INOUT=1
      RETURN
    3 INOUT = 1
      IF(TEMP1.LT.0.0) INOUT = 0
      RETURN
    4 INOUT = 1
      IF(TEMP1/TEMP3.GT.0.0) INOUT=0
      IF(TEMP2/TEMP3.LT.0.0) INOUT=0
      RETURN
      END
```

Appendix B

SERVICING-RATE PROGRAM EXAMPLE

The TIMER program of App. A creates an output intervisibility file containing, for each observer/route combination, a table whose place positions represent consecutive equal-length segments (usually 25 m) along the route and whose entries are the ranges from the observer to the centers of the route segments if the segments are visible from the observer and zero otherwise.

This appendix contains a description and program listing for a computer routine that uses the TIMER output intervisibility file as an input. This model calculates the servicing rate (see Sec. IV) for firing units that are in defensive positions at the observer locations and companies of enemy tanks approaching along the routes.

Table B.1 presents the input formats and Table B.2 presents an example of the servicing-rate program FORTRAN listing.

Table B.1

INPUT FORMATS

Card Columns	Data Definitions
Card 1	
1-10	Number of observers
11-20	Number of routes
21-30	Number of route segments
Card 2	
1-10	Tank company velocity (km/hr)
11-20	Maximum number of rounds fired at each tank
Card 3	
1-10	Number of rounds for firing unit type 1
11-20	Number of rounds for firing unit type 2
21-30	Number of rounds for firing unit type 3
31-40	Number of rounds for firing unit type 4
Card 4	
1-10	Minimum range (km) for firing unit type 1
11-20	Minimum range (km) for firing unit type 2
21-30	Minimum range (km) for firing unit type 3
31-40	Minimum range (km) for firing unit type 4
Card 5	Same as card 4 for maximum range
Card 6	
1-10	Time (sec) to fire one round for firing unit type 1
11-20	Time (sec) to fire one round for firing unit type 2
21-30	Time (sec) to fire one round for firing unit type 3
31-40	Time (sec) to fire one round for firing unit type 4
Card 7	
1-10	Firing interval (sec) before moving for firing unit type 1
11-20	Firing interval (sec) before moving for firing unit type 2
21-30	Firing interval (sec) before moving for firing unit type 3
31-40	Firing interval (sec) before moving for firing unit type 4
Card 8	
1-10	Movement interval (sec) for firing unit type 1
11-20	Movement interval (sec) for firing unit type 2
21-30	Movement interval (sec) for firing unit type 3
31-40	Movement interval (sec) for firing unit type 4

Card Columns	Data Definitions
Card 9	
1-5	Starting segment for tank company on route 1
...	
71-75	Starting segment for tank company on route 15
Card 10	
1-1	Type (1 to 4) of firing unit at observer location 1
...	
80-80	Type (1 to 4) of firing unit at observer location 80
Card 11	
1-1	Type (1 to 4) of firing unit at observer location 81
...	
22-22	Type (1 to 4) of firing unit at observer location 102
Card 12	
1-4	Four character name for route 1
...	
57-60	Four character name for route 15
Card 13	
1-4	Four character name for observer 1
...	
80-80	Four character name for observer 20
Card 14-18	
	Continue names for observers 21 to 102

Table B.2

SERVICING-RATE PROGRAM FORTRAN LISTING EXAMPLE

```
C     MAIN
      INTEGER*2 MINRGE(4),MAXRGE(4),FRATE(4),FTIME(4),MOVE(4),ROUNDS(4)
      INTEGER*2 START(102),FIRST(102),LAST(102),SHOT(102)
      INTEGER*2 TYPE (102),ITOTAL(102),F(102,15),FF(102,15)
      INTEGER*2 INIT(15),C(15,10),CC(15,10),P(15,10),JTOTAL(15)
      INTEGER X(102,15,27),MASK(32)
      INTEGER*2 MARK(800)
      DATA MASK/Z1,Z2,Z4,Z8,Z10,Z20,Z40,Z80,Z100,Z200,Z400,Z800,Z1000,
     1 Z2000,Z4000,Z8000,Z10000,Z20000,Z40000,Z80000,Z100000,Z200000,
     2 Z400000,Z800000,Z1000000,Z2000000,Z4000000,Z8000000,Z10000000,
     3 Z20000000,Z40000000,Z80000000/
      DIMENSION A(15),B(102)
      DATA A/15*'    '/
C
C EXAMPLE SERVICING RATE PROGRAM
C
      READ(5,100) NPOS,NROUTE,NSEG
  100 FORMAT(3I10)
      READ(5,101) VEL,MXSHOT,ROUNDS,MINRGE,MAXRGE,FRATE,FTIME,MOVE
  101 FORMAT(F10.0,I10/(4I10))
      WRITE(6,102) VEL,MXSHOT,ROUNDS,MINRGE,MAXRGE,FRATE,FTIME,MOVE
  102 FORMAT('1','VELOCITY =',F5.0,' KM/HR'/
     * '0','MAXIMUM ROUNDS PER TARGET =',I5/
     * '0',23X,' M60A1 ',9X,'M551 ',10X,'TOW',8X,'DRAGON'/
     * '0','NO. OF ROUNDS:      ',4(I10,'      ')/
     * '0','MIN FIRING RANGE: ',4(I10,' M   ')/
     * ' ','MAX FIRING RANGE: ',4(I10,' M   ')/
     * ' ','FIRING RATE: 1 PER',4(I10,' SEC')/
     * ' ','FIRING INTERVAL:  ',4(I10,' SEC')/
     * ' ','MOVEMENT INTERVAL:',4(I10,' SEC'))
      READ(5,103) (INIT(N),N=1,NROUTE)
  103 FORMAT(15I5)
      WRITE(6,104) (INIT(N),N=1,NROUTE)
  104 FORMAT('0','STARTING POINTS ON ROUTES: ',15I5)
      READ(5,105) (TYPE(N),N=1,NPOS)
  105 FORMAT(80I1)
      WRITE(6,106) (TYPE(N),N=1,NPOS)
  106 FORMAT('0','WEAPON TYPES: ',102I1)
C A(N) IS A FOUR CHARACTER NAME FOR ROUTE N
C B(N) IS A FOUR CHARACTER NAME FOR POSITION N
      READ(5,107) (A(N),N=1,NROUTE)
      READ(5,107) (B(N),N=1,NPOS)
  107 FORMAT(20A4)
C V IS VELOCITY IN METERS/SECOND
```

```
      V=VEL*1000/3600
C INTIALIZE COUNTER TABLES
C C(I,J) IS CUMULATIVE ROUNDS ASSIGNED TO TANK J OF ROUTE I
C CC(I,J) IS CUM. ROUNDS ASSIGNED TO TANK J ROUTE I THRU PREVIOUS MINUTE
C F(I,J) IS CUM. ROUNDS ASSIGNED TO ROUTE J FROM POSITION I
C FF(I,J) IS CUM. ROUNDS ASSIGNED TO ROUTE J FROM I THRU PREVIOUS MINUTE
      DO 1 I=1,NROUTE
      DO 1 J=1,10
      C(I,J)=0
    1 CC(I,J)=0
      DO 2 I=1,NPOS
      DO 2 J=1,NROUTE
      F(I,J)=0
    2 FF(I,J)=0
C
C READ IN OBSERVATION POINT/ROUTE SEGMENT INTERVISIBILITY ARRAYS AND
C USE BINARY INDICATORS FOR INDICATING SEGMENTS IN VIEW AND IN RANGE
C
      DO 5 J=1,NROUTE
      DO 5 M=1,NPOS
      RMIN=MINRGE(TYPE(M))
      RMAX=MAXRGE(TYPE(M))
      DO 3 K=1,26
    3 X(M,J,K)=0
      READ(1,110) (MARK(JK),JK=1,NSEG)
  110 FORMAT(20(40A2))
      DO 4 K=1,NSEG
      IF(MARK(K).LE.RMIN.OR.MARK(K).GT.RMAX) GO TO 4
      JJ=(K-1)/30+1
      I=K-30*(JJ-1)
      X(M,J,JJ)=X(M,J,JJ)+MASK(I)
    4 CONTINUE
    5 CONTINUE
C
C INITIALIZE EVENT TIME INDICATORS FOR EACH OBSERVATION POINT
C FOR EACH FIRING UNIT:
C START IS THE TIME AT WHICH THE FIRING UNIT WILL NEXT BE READY TO FIRE
C FIRST IS THE TIME AT WHICH THE FIRST ROUND WAS FIRED AFTER MOVING
C LAST IS THE TIME AT WHICH THE MOST RECENT ROUND WAS FIRED
C SHOT IS THE NUMBER OF ROUNDS REMAINING
C
C N1 IS THE TIME STEP (SEC)
C
      N1=10
      DO 6 M=1,NPOS
      SHOT(M)=ROUNDS(TYPE(M))
      START(M)=N1
      FIRST(M)=-1000
    6 LAST(M)=-1000
C
```

```
C REPEAT CALCULATIONS FOR EACH TIME STEP UNTIL SOME TANK UNIT REACHES
C THE END OF ITS ROUTE
C
      DO 90 N=N1,1500,N1
      NN=N/60
      DO 30 M=1,NPOS
      IF(SHOT(M).LE.0) GO TO 30
      ITYPE=TYPE(M)
      IF(FIRST(M)+FTIME(ITYPE)-FRATE(ITYPE).LT.N)
     *             START(M)=FIRST(M)+FTIME(ITYPE)+MOVE(ITYPE)
      IF(START(M).GT.N) GO TO 30
      IF(LAST(M)+FRATE(ITYPE).GT.N) GO TO 30
C DETERMINE ALL TANK POSITIONS THAT CAN BE FIRED UPON FROM POSITION N
C IFLAG NOT EQUAL TO ZERO INDICATES  AT LEAST ONE FIRING OPPORTUNITY
C FOR THE FIRING UNIT THIS TIME PERIOD
      IFLAG=0
      DO 15 I=1,NROUTE
      INITX=INIT(I)
      DO 15 J=1,10
C P(I,J) IS THE CUM. NUMBER OF ROUNDS FIRED ON TANK J OF ROUTE I
C DEFAULT VALUE OF 10000 INDICATES TANK CANNOT BE FIRED UPON THIS PERIOD
      P(I,J)=10000
      IF(C(I,J).EQ.MXSHOT) GO TO 15
      IV=V*N/25-(J-1)*2 + INITX
      IF(IV.LE.NSEG) GO TO 11
      WRITE(6,115)
  115 FORMAT('1',' THE PROGRAM HAS RUN OUT OF ROUTE SEGMENTS')
      CALL EXIT
C DETERMINE IF TANK J OF ROUTE I IS VISIBLE AND WITHIN RANGE FOR ENTIRE
C  ROUND FIRING TIME
   11 IF(IV.LE.INITX) GO TO 15
      ID=(IV-1)/30+1
      MK=X(M,I,ID)
      IF(MK.EQ.0) GO TO 15
      ID=IV-30*(ID-1)
      MK=MK-MASK(ID)-MASK(ID+1))*(MK/MASK(ID+1))
      IF(MK.LT.0) GO TO 15
      P(I,J)=C(I,J)
      IVV=FRATE(ITYPE)*V/25.0+0.5-1.0
      IF(IVV.LE.0) GO TO 14
      DO 13 K=1,IVV
      IV=IV-1
      IF(IV.LE.0) GO TO 12
      ID=(IV-1)/30+1
      MK=X(M,I,ID)
      IF(MK.EQ.0) GO TO 12
      ID=IV-30*(ID-1)
      MK=MK-MASK(ID)-MASK(ID+1))*(MK/MASK(ID+1))
      IF(MK.NE.0) GO TO 13
   12 P(I,J)=10000
```

```
      GO TO 15
   13 CONTINUE
   14 IFLAG=1
   15 CONTINUE
C SELECT AS TARGET THAT TANK WITH THE FEWEST NUMBER OF ROUNDS ASSIGNED
      IF(IFLAG.EQ.0) GO TO 30
      MINP=10000
      DO 20 I=1,NROUTE
      DO 20 J=1,10
      IF(P(I,J).GE.MINP) GO TO 20
      MINP=P(I,J)
      II=I
      JJ=J
   20 CONTINUE
      IF(MINP.EQ.10000) GO TO 30
      C(II,JJ)=C(II,JJ)+1
      F(M,II)=F(M,II)+1
      LAST(M)=N
      SHOT(M)=SHOT(M)-1
      IF(FIRST(M).LT.START(M)) FIRST(M)=N
   30 CONTINUE
C
C AT END OF EACH MINUTE PRINT TABLES SHOWING NUMBER OF ROUNDS ASSIGNED
C
C PRINT ROUNDS ASSIGNED TO EACH TANK THIS MINUTE, BY ROUTE
      IF(MOD(N,60).NE.0) GO TO 90
      NTOTAL=0
      DO 41 I=1,NROUTE
      JTOTAL(I)=0
      DO 41 J=1,10
      CC(I,J)=C(I,J)-CC(I,J)
      JTOTAL(I)=JTOTAL(I)+CC(I,J)
      NTOTAL=NTOTAL+CC(I,J)
   41 CONTINUE
      WRITE(6,120) NN,(I,I=1,10)
  120 FORMAT('1',' TIME =',I3,' MIN',50X,'TARGET #'/
     * '0',15X,10I10,10X,'TOTALS'/)
      DO 42 I=1,NROUTE
   42 WRITE(6,121) I,A(I),(CC(I,J),J=1,10),JTOTAL(I)
  121 FORMAT(' ',I5, 2X,A4,4X,10I10,I15)
      WRITE(6,122) NTOTAL
  122 FORMAT('0',115X,I15)
C PRINT TOTAL NUMBER OF TARGETS SERVICED THIS MINUTE
      KTANK=0
      DO 43 I=1,NROUTE
      DO 43 J=1,10
      IF(CC(I,J).NE.0) KTANK=KTANK+1
   43 CC(I,J)=C(I,J)
C PRINT NUMBER OF TANKS SERVICED THIS MINUTE
      WRITE(6,123) KTANK
```

```
  123 FORMAT('0',20X, 'NUMBER OF TARGETS SERVICED THIS MINUTE =',I4)
C PRINT ROUNDS ASSIGNED TO EACH ROUTE THIS MINUTE, BY POSITION
      DO 45 I=1,NPOS
      ITOTAL(I)=0
      DO 45 J=1,NROUTE
      FF(I,J)=F(I,J)-FF(I,J)
   45 ITOTAL(I)=ITOTAL(I)+FF(I,J)
      WRITE(6,125) NN,(A(I),I=1,15),(J,J=1,NROUTE)
  125 FORMAT('0',' TIME =',I3,' MIN',50X,'ROUTES'/' ',15X,15(2X,A4)/
     * ' ',15X,15I6/)
      WRITE(6,126)
  126 FORMAT('+',110X,'TOTALS',6X,'CUM.TOTAL')
      NTOTAL=0
      DO 46 I=1,NPOS
      NTOTAL=NTOTAL+ITOTAL(I)
      IF(ITOTAL(I).EQ.0) GO TO 46
      WRITE(6,127) I,B(I),(FF(I,J),J=1,NROUTE)
  127 FORMAT(' ',I3,2X,A4,6X,15I6)
      WRITE(6,128) ITOTAL(I),NTOTAL
  128 FORMAT('+',105X,I10,I15)
   46 CONTINUE
      WRITE(6,129) (JTOTAL(J),J=1,NROUTE)
  129 FORMAT('0',15X,15I6)
      DO 47 I=1,NPOS
      DO 47 J=1,NROUTE
   47 FF(I,J)=F(I,J)
C CUMULATIVE ROUNDS ASSIGNED TO EACH TANK, BY ROUTE
      NTOTAL=0
      DO 51 I=1,NROUTE
      JTOTAL(I)=0
      DO 51 J=1,10
      JTOTAL(I)=JTOTAL(I)+C(I,J)
      NTOTAL=NTOTAL+C(I,J)
   51 CONTINUE
      WRITE(6,130) NN,(I,I=1,10)
  130 FORMAT('0',' CUMULATIVE RESULTS'/' ',' TIME =',I3,' MIN',50X,
     * 'TARGET #'/'0',15X,10I10,10X,'TOTALS'/)
      DO 52 I=1,NROUTE
   52 WRITE(6,131) I,A(I),(C(I,J),J=1,10),JTOTAL(I)
  131 FORMAT(' ',I5, 2X,A4,4X,10I10,I15)
      WRITE(6,132) NTOTAL
  132 FORMAT('0',115X,I15)
C PRINT CUMULATIVE NUMBER OF TANKS SERVICED
      KTANK=0
      KTANK2=0
      KTANK3=0
      DO 53 I=1,NROUTE
      DO 53 J=1,10
      IF(C(I,J).EQ.1) KTANK2=KTANK2+1
      IF(C(I,J).GE.2) KTANK2=KTANK2+2
```

```
      IF(C(I,J).EQ.1) KTANK3=KTANK3+1
      IF(C(I,J).EQ.2) KTANK3=KTANK3+2
      IF(C(I,J).GE.3) KTANK3=KTANK3+3
   53 IF(C(I,J).NE.0) KTANK=KTANK+1
      WRITE(6,133) KTANK,KTANK2,KTANK3
  133 FORMAT('0',20X, 'NUMBER OF TARGETS SERVICED THRU THIS MINUTE =',
     * I4/' ',13X,'NUMBER OF (PAIRED) TARGETS SERVICED THRU THIS MINUTE
     *=',I4/' ',13X,'NUMBER OF (TRIPLED) TARGETS SERVICED THRU THIS MINU
     *TE =' ,I4)
C PRINT CUMULATIVE ROUNDS ASSIGNED TO EACH ROUTE, BY POSITION
      DO 55 I=1,NPOS
      ITOTAL(I)=0
      DO 55 J=1,NROUTE
   55 ITOTAL(I)=ITOTAL(I)+F(I,J)
      WRITE(6,135) NN,(A(I),I=1,15),(J,J=1,NROUTE)
  135 FORMAT('0',' CUMULATIVE RESULTS'/' ',' TIME =',I3,' MIN',50X,
     * 'ROUTES'/' ',15X,15(2X,A4)/' ',15X,15I6/)
      WRITE(6,126)
      NTOTAL=0
      DO 56 I=1,NPOS
      NTOTAL=NTOTAL+ITOTAL(I)
      IF(ITOTAL(I).EQ.0) GO TO 56
      WRITE(6,127) I,B(I),(F(I,J),J=1,NROUTE)
      WRITE(6,128) ITOTAL(I),NTOTAL
   56 CONTINUE
      WRITE(6,137) (JTOTAL(J),J=1,NROUTE)
  137 FORMAT('0',15X,15I6)
C
   90 CONTINUE
      END
```